EXPLORATION & ANALYSIS OF OSM WHEELCHAIR HISTORY & MICROM DATA

Aadesh Misra

A thesis submitted for the degree of
Master of Science

June 2016

Preface

This document is the Masters thesis document required to complete the Masters programme at *Technische Hochschule(Technical University of Applied Sciences), Cologne* in collaboration with *Fraunhofer IAIS, Sankt-Augustin.* The thesis is under the guidance of Dr. Hans Voss at the Fraunhofer Institute and Dr. Heide Faeskorn-Woyke at TH, Cologne. The topic of the thesis falls under the business domain of social intelligence and under the technical domain of data analysis. The document discusses the origin, methodology, analysis and results obtained during the course of the thesis.

Contents

Chapter 1

Introduction

1.1 Social Intelligence

Social intelligence is an aggregated measure of self- and social-awareness, evolved social beliefs and attitudes, and a capacity and appetite to manage complex social change.[12]

Social scientists and policy makers face the challenge when dealing with unquantifiable or highly variable entities, for example, attitudes, beliefs, awareness, etc. Data and visualization-driven strategies provide useful insight into the nature of relationships both complex and simple gathered from various sources. The relationships are used by social scientists for motivation, fund raising and solution building strategies. The recent interest of public and commercial entities towards the integration of people with disabilities into the mainstream has increased for different reasons. For example, smartphones come with built in *Accessibility features*[31] like talkback. The construction of cities and buildings are also being evaluated on the accessibility criteria.

The integration of people with disabilities in the society is a challenge. The people with single and multiple disabilities is significantly high both in number and percentage. Around 15% of the world's population, or estimated 1 billion people, live with disabilities. Therefore, making them the world's largest neglected minority. Despite the numbers, people with disabilities are neglected and ignored. A more significant effort needs to be made to convert the people with disabilities into a community. Thus, technology can play an extremely important role. Unfortunately, existing technologies also have less or no focus on the needs of the differently-abled. The thesis utilizes data gathered as a part of *Collectively improving accessibility in European cities(CAP4Access)* project and explore some of the interesting patterns for

further motivation.

1.2 About Cap4Access Project

Whenever the people on wheelchair plan to access a toilet, a building, any public place or just travel from point A to B just to reach a place, it often becomes a challenge for them. More than the distance or inaccessibility, what pains them more is the loss of self-respect. Technology can play a key role to achieve this goal. A study by *Macdonald and Clayton*[32] provides an insight into the existing and widening digital divide between the normal users and the users with special requirements. A probable reason is that the current technology is focused on the normal users and the accessibility is just seen as an overhead. The *CAP4Access* project is an initiative from the European Commission to integrate the people on wheelchair or with other movement disabilities into the mainstream society.

The objective of CAP4Access is to develop and pilot-test methods and tools for collectively gathering and sharing information about the accessibility of public spaces.[11] During the course of the project, the data collection mainly takes place through crowd sourcing campaigns. The information about campaigns can be found at the link: http://myaccessible.eu/campaigns. The users mark the various public places like toilets, shops, bus stops, etc as one of the three categories i.e. *yes* in case its accessible, *no* in case its not accessible, *limited* in case it was partially accessible, the left out point of interests also known as POIs, are by default included under the *unknown* category.

The initiative utilizes the domain expertise of various reputed institutions from Europe to develop solutions to achieve the goal of *Accessible Europe*. The teams from these institutions develop their use cases giving primary importance to problems of people with movement disabilities. The list of partners involved can be found at the link: http://cap4access.eu/team.

Fraunhofer IAIS located in Sankt-Augustin, Germany has the responsibility to collect and integrate data, and also develop a dashboard with informative visualizations to communicate the results. The project initially utilized the database maintained by OpenstreetMap(OSM) community but issues like low performance, custom data requirements and inadequate support from the OSM community led to the creation of a localized database for the project. The visualizations are both statistical and geographical in nature. The development is done using NodeJS, AngularJS framework, Openlayers3 map library and a charts API called c3.js (an implementation of D3.js). The dashboard is available at http://kd-cap4access.iais.fraunhofer.de with username as *c4a* and password *cap4access*.

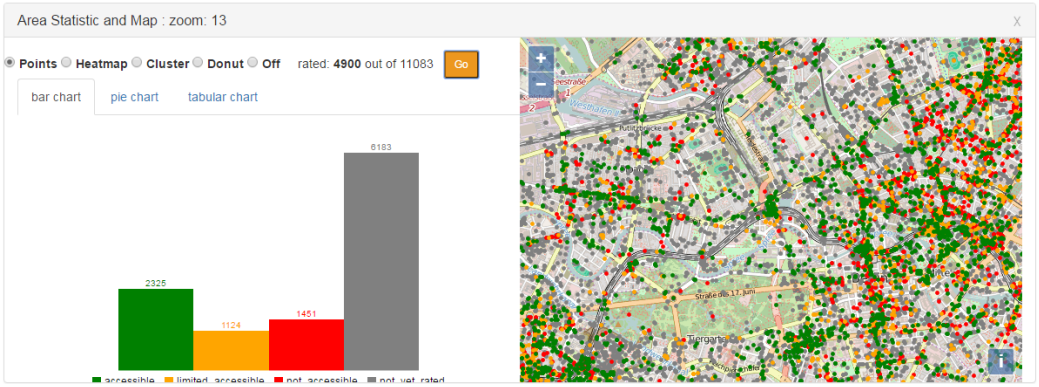

Figure 1.1: Sample visualization displaying histogram and *Points of Interest* on the map.

1.3 Data Sources

The thesis is provided with two data sources, the first is *OSM Wheelchair History*(OWH) and the second is *MICROM*. While *MICROM* is a data store containing multiple tables, the *OSM Wheelchair History* is a single table storing historical data about nodes, thus we categorize both as data sources and not as data sets. These data sources provide the fuel in the form of datasets to the analysis. The two data sources and their corresponding nature is discussed in the following subsections.

1.3.1 OSM Wheelchair History

OpenStreetMap Foundation provides open source and freely usable geo-spatial data. *OpenStreetmap's aim is to create a set of map data that is free to use, editable, and licensed under new copyright schemes.*[19] A localized filtered subset of OSM database has been created and is being maintained as a part of *CAP4Access* initiative by Fraunhofer IAIS, Sankt-Augustin. The database technologies utilized are Oracle RDBMS and PostgreSQL. Figure 1.2 shows the complete description of the *OWH* table with comments.

A *NODE* is a point in space defined by its *ID*, *longitudinal* and *latitudinal* values. *NODEID* is the unique *ID* that identifies each node. There may be a case where the *NODE* changes its description, for example, its location which is stored in the variables *LON* and *LAT*, the new changes are then added with an updated *VERSION* provided by OSM and an updated *HISTORY_SEQ* is generated as a part of the *CAP4Access* project and has the same function as the *VERSION* variable but without any gaps. The accessibility for each *NODE* is contained in the variable *WHEELCHAIR_VALID* and it can take

one of the four values i.e. *yes*, *no*, *limited* or *unknown*. Other variables like *NAME, SHOP, AMENITY, etc* store the information about the type of *NODE*. The variables discussed above form the backbone of the *OWH* data source.

1.3.2 MICROM

MICROM is an organization that collects data in the domain of *consumer marketing*. To learn more about the organization visit http://www.microm-online.de/home. It is a collection of various tables that contain the geo-demographic data for Germany. The tables contain data grouped by *POSTAL CODE(PLZ), MUNICIPALITIES(GKZ)* and *NAVTEQ links(LINK_ID)*. The important tables that are part of the *MICROM* data source are:

1. ***dp_postleitzahl*** : The table includes data relevant to the postal area. For example postal code(*PLZ*) and the geometry of the postal area.

2. ***dp_gemeinde*** : The table includes data relevant to the municipal area. For example, the municipal code(*GKZ*), municipality name and the geometry of the municipal area.

3. ***plz_eag*** : The table includes the data regarding the distribution of population with respect to the age group and gender in a postal area. For example total number of men between the age group of 0 to 3 years.

4. ***plz_kkr*** : The table includes financial data about the people in the postal area. For example the average purchasing power per household.

5. ***plz_mba*** : The table includes data like number of households, number of commercial enterprises and the total number of houses in the postal area.

6. ***plz_mbe_gebaeudetyp*** : The table includes the data regarding the number of commercial and privately used buildings in a postal area.

7. ***plz_mso_status*** : The table contains data about the number of low, average and high-status households in a postal area.

1.3.3 Integration Of Data Sources

The *OSM Wheelchair History* data source provides information about the tagging activity but lacks the socio-demographic description about the region where the tag exists. *MICROM* data source has a number of tables

	Name	Type	Nullable	Default	Comments
1	NODEID	NUMBER			[att] id of the OSM element, positive for nodes and negative for ways. Inacurate name is keept for historical reasons!
2	VERSION	NUMBER			[att] version of the OSM element
3	HISTORY_SEQ	NUMBER			[att] sequence within OWH, starts with 1, no gaps
4	IS_NODE	NUMBER			[att] =1 iff object is a node (and id>0); =0 iff object is a way (and id<0)
5	CHANGESET	NUMBER			[att] id of the changeset
6	TIMESTAMP	DATE			[att] timestamp of the OSM element
7	UID_	NUMBER	Y		[att] id of the user who edited this version
8	USER_	VARCHAR2(112 CHAR)	Y		[att] name of the user who edited this version
9	VISIBLE	NUMBER(1)			[att] 1 iff version is visible, 0 iff version is not visible
10	LAST_VERSION_VISIBLE	NUMBER(1)			[flag-node] indicates if the last version of the node is visible
11	LAST_VISIBLE	NUMBER(1)			[flag-vers] indicates if this version of the node is the last visible revision
12	LAST_VIS_IN_CHANGESET	NUMBER(1)			[flag-vers] indicates if this version of the node is the last visible revision within the corresponding changeset
13	WHEELCHAIR_NODE	NUMBER(1)			[flag-node] indicates if any version of the node has a wheelchair-tag
14	VALID_WHEELCHAIR_NODE	NUMBER(1)			[flag-node] indicates if any version of the node has a valid wheelchair-tag [lower(wheelchair) in ('yes','no','limited')]
15	WHEELMAP_POI_NODE	NUMBER(1)			[flag-node] indicates if any version of the node falls within the WheelMap-POI-filter
16	OTHER_POI_NODE	NUMBER(1)			[flag-node] indicates if any version of the node falls within the rest of OWH-poi-filter
17	WHEELCHAIR_VERSION	NUMBER(1)			[flag-vers] indicates if this version has a wheelchair-tag
18	VALID_WHEELCHAIR_VERSION	NUMBER(1)			[flag-vers] indicates if this version has a valid wheelchair-tag [lower(wheelchair) in ('yes','no','limited')]
19	WHEELMAP_POI_VERSION	NUMBER(1)			[flag-vers] indicates if this version falls within the WheelMap-POI-filter
20	OTHER_POI_VERSION	NUMBER(1)			[flag-vers] indicates if this version falls within the rest of OWH-poi-filter
21	WHEELCHAIR	VARCHAR2(254)	Y		[tag] value of wheelchair-tag of the version
22	WHEELCHAIR_VALID	VARCHAR2(7)	Y		[tag] if lower(wheelchair) in ('yes','no','limited') then lower(wheelchair) else NULL
23	WC_VALID_STATUS_CHANGE	NUMBER(1)			[flag-vers] indicates if there is a change in wheelchair_valid on this version; but only on visible nodes
24	WC_VALID_STATUS_DURATION	NUMBER	Y		[value-vers] iff wc_valid_status_change=1 then duration of the wheelchair_valid in seconds (NULL for last one); otherwise NULL
25	WHEELMAP_POI_TAGS	VARCHAR2(97)	Y		[tag-agg] concatenation of all tags that fall within the Wheelmap-POI-filter
26	OTHER_POI_TAGS	VARCHAR2(86)	Y		[tag-agg] concatenation of all tags that fall within the rest of OWH-poi-filter
27	NAME	VARCHAR2(255 CHAR)	Y		[tag] value of name-tag of the version
28	TOILETS_WHEELCHAIR	VARCHAR2(46 CHAR)	Y		[tag] value of toilets:wheelchair-tag of the version
29	WHEELCHAIR_DESCRIPTION	VARCHAR2(258 CHAR)	Y		[tag] value of wheelchair:description-tag of the version
30	WHEELCHAIR_DESCRIPTION_EN	VARCHAR2(254)	Y		[tag] value of wheelchair:description:en-tag of the version
31	CAPACITY	VARCHAR2(219 CHAR)	Y		[tag] value of capacity-tag of the version
32	CAPACITY_DISABLED	VARCHAR2(39)	Y		[tag] value of capacity:disabled-tag of the version
33	WHEELCHAIR_ENTRANCE_WIDTH	VARCHAR2(9)	Y		[tag] value of wheelchair:entrance_width-tag of the version
34	WHEELCHAIR_STEP_HEIGHT	VARCHAR2(5)	Y		[tag] value of wheelchair:step_height-tag of the version
35	WHEELCHAIR_PLACES	VARCHAR2(22)	Y		[tag] value of wheelchair:places-tag of the version
36	RAMP_WHEELCHAIR	VARCHAR2(7)	Y		[tag] value of ramp:wheelchair-tag of the version
37	SHOP	VARCHAR2(126 CHAR)	Y		[tag] value of shop-tag of the version
38	AMENITY	VARCHAR2(255)	Y		[tag] value of amenity-tag of the version
39	PUBLIC_TRANSPORT	VARCHAR2(152)	Y		[tag] value of public_transport-tag of the version
40	HIGHWAY	VARCHAR2(189)	Y		[tag] value of highway-tag of the version
41	RAILWAY	VARCHAR2(139)	Y		[tag] value of railway-tag of the version
42	AERIALWAY	VARCHAR2(30)	Y		[tag] value of aerialway-tag of the version
43	AEROWAY	VARCHAR2(39 CHAR)	Y		[tag] value of aeroway-tag of the version
44	NATURAL	VARCHAR2(70 CHAR)	Y		[tag] value of natural-tag of the version
45	BUILDING	VARCHAR2(255)	Y		[tag] value of building-tag of the version
46	OFFICE	VARCHAR2(183 CHAR)	Y		[tag] value of office-tag of the version
47	TOURISM	VARCHAR2(254 CHAR)	Y		[tag] value of tourism-tag of the version
48	LEISURE	VARCHAR2(70 CHAR)	Y		[tag] value of leisure-tag of the version
49	HISTORIC	VARCHAR2(133)	Y		[tag] value of historic-tag of the version
50	SPORT	VARCHAR2(255)	Y		[tag] value of sport-tag of the version
51	LON	NUMBER	Y		[att] longitude (wgs84) of the version
52	LAT	NUMBER	Y		[att] latitude (wgs84) of the version
53	LAST_VISIBLE_LON	NUMBER			[att-node] longitude (wgs84) of the last visible version of the node
54	LAST_VISIBLE_LAT	NUMBER			[att-node] latitude (wgs84) of the last visible version of the node

Figure 1.2: Description Of The OWH Table

that contain socio-demographic variables which provide information regarding the postal area, municipal area, total population, sex ratio, etc. Thus, integration of *OSM Wheelchair History* and *MICROM* data sources link together a host of relevant variables which help describe each *Point of Interest* and its surrounding region. The data sources are compatible for integration as they contain geographical variables like longitude, latitude, and geometries. Multiple data sets can be generated from the resulting integrated data source for various analytical purposes. For example, to find the cities or postal area with high tagging activity or to find the average per capita income for a household in the postal code where the *POI* is marked. Thus, the integration of the data sources provides high value for analytics.

1.4 Objective Of The Thesis

The thesis combines three domains i.e. computer science, data analytic and social intelligence. The focus of the thesis is to utilize computer science and analytic methodologies to explore and answer the questions relevant to the domain of social intelligence. The nature of the thesis is explorative, therefore, the variables from the combined data sources help explore and develop various social intelligence questions relevant to the *CAP4Access* project.

The following statements form the core objectives of the thesis:

1. Identification of trends and patterns in the tagging activity.
2. Exploration of variables to analyse the tagging activity.
3. Application of supervised machine learning algorithms to classify/predict if the *POI* is tagged or not. Also, extend the analysis to classify/predict in case the *POI* is tagged and the *POI* is tagged as *yes*, *no* or *limited*.
4. Application of unsupervised clustering algorithms to identify density based clusters.
5. Application of unsupervised association rule algorithms to identify frequent sets in the data.

Chapter 2

Methodology

2.1 Data Exploration Methodology

An appropriate methodology is required for the task of data exploration as it provides a standardized framework and methods to efficiently extract knowledge from the data. There are three popular methodologies in the community for data exploration tasks. Knowledge discovery in databases (KDD) as discussed in the research paper by *Fayyad*[13], is one of the popular data mining methodologies as shown in Figure 2.1. The methodology is iterative and interactive with user decision oriented at each step.

Another popular methodology provided by SAS called SEMMA (Sample, Explore, Modify, Model, Assess), provides a data mining methodology for extracting knowledge from the database as shown in Figure 2.2. SAS is a market leader in the field of data analytics and has a rich set of analytical tools which are modelled towards the same methodology.

Cross Industry Standard Process for Data Mining (CRISP-DM) has by far become the most popular methodology because it includes the business understanding as its first step. Understanding the domain helps formulate questions or objectives the user is looking for in the data, this makes him/her extremely efficient. Figure 2.3 gives an overview of the methodology.

Although databases are the prominent source of data, yet they may not be the only source of data. For example, web scrapping or integration of live feeds may also be relevant sources of data. Another parameter for methodology selection is the concept of question formation or objective selection. Formation of relevant questions, streamline the process of explorative analysis. *KDD* is built upon only keeping in mind a singular relational database as an analysis source, which although appropriate might limit the scope of future implementations. *KDD* also lacks thought and application of busi-

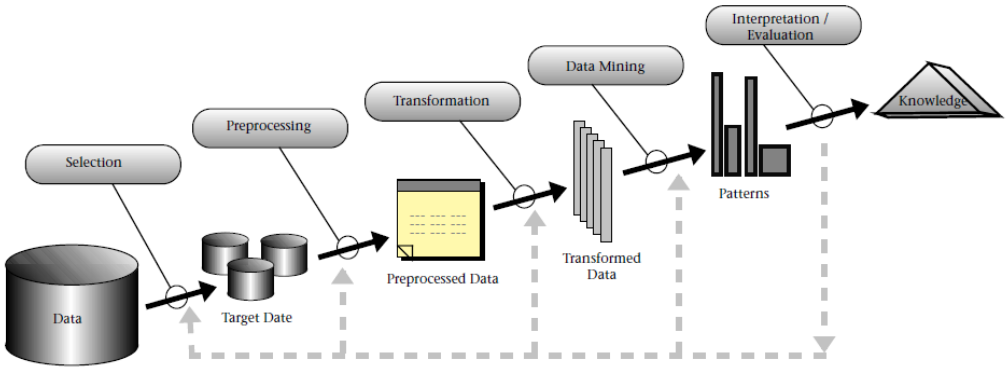

Figure 2.1: An Overview Of The Steps That Compose The KDD Process.

ness understanding. *SEMMA* is similar to *KDD* except when implemented using SAS infrastructure it enables users to integrate multiple data sources and perform efficient data analysis. Due to license issues with SAS tools and infrastructure, it may not be an appropriate option, although it makes the life of user extremely simple and helps him/her focus only on the task at hand. *CRISP-DM* fits the needs as it starts with business understanding, which allows for revolving around relevant data, focusing equally on data understanding resulting in appropriate data preparation to apply various modelling methods. *CRISP-DM*, unlike SEMMA can unambiguously be applied irrespective of the tools.

2.2 Thesis Methodology

The practical methodology of the thesis utilizes *CRISP-DM* as the base framework. It includes understanding the socio-economic domain which is relevant to the thesis. Integration of the two data sets provided i.e. *OSM Wheelmap history* and *MICROM*. The practical methodology applied can be viewed in Figure 2.4. The practical methodology has six steps :

1. Formation of a relevant question to be answered.

2. Dataset generation from the integrated data source.

3. Process and transform the data according to the algorithm or tools being used.

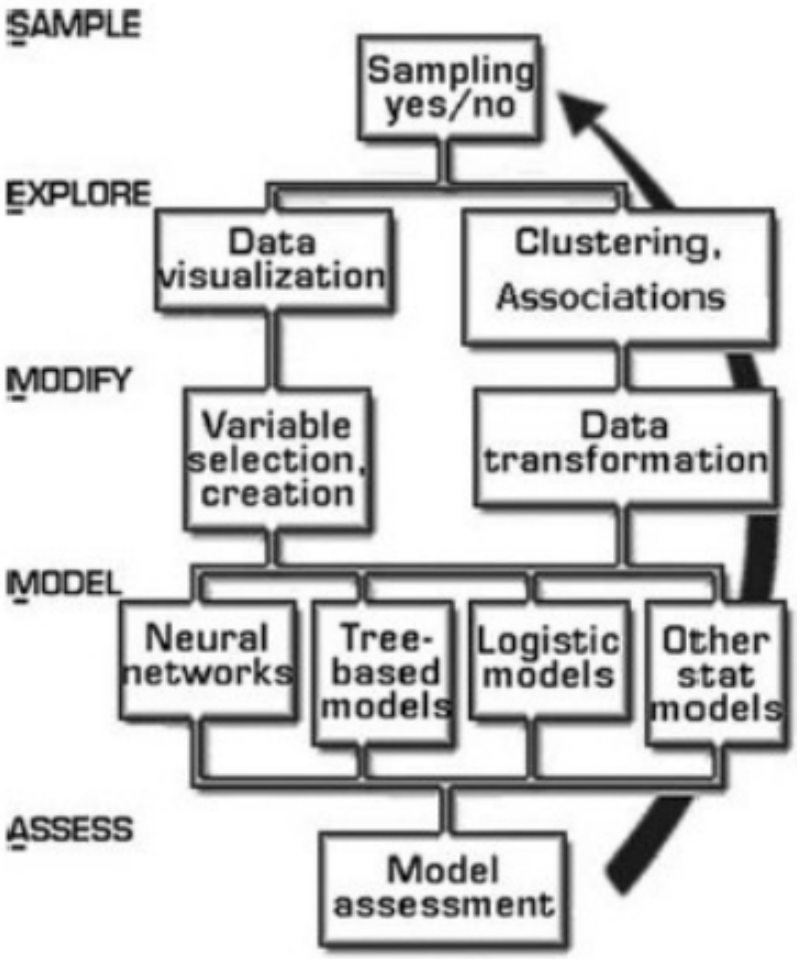

Figure 2.2: SEMMA Methodology From SAS

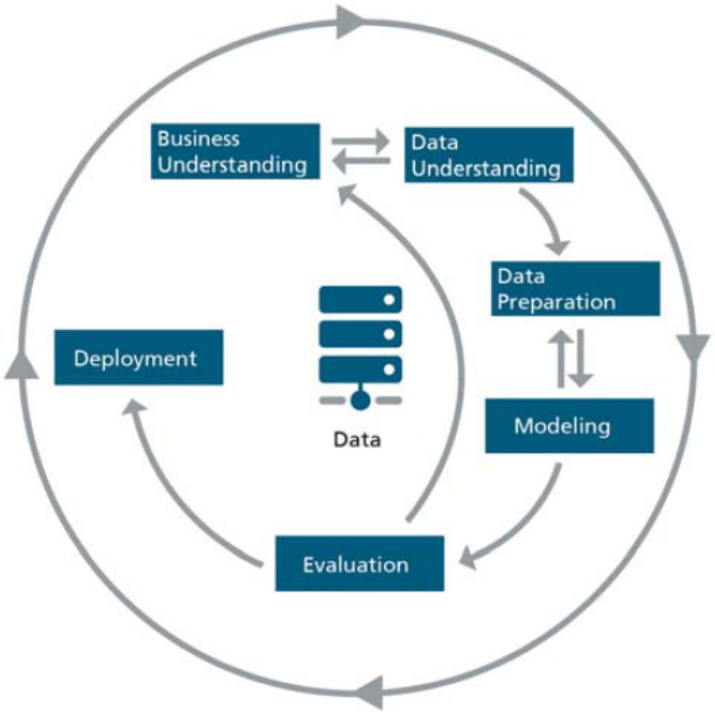

Figure 2.3: Cross Industry Standard Process for Data Mining (CRISP) - Process Overview

4. Application of algorithms or tools to obtain the result.

5. Evaluation of results by chosen metrics or standards. If the results do not conform to the evaluation metric then move to 4 with new parameters. If the results do not conform to the evaluation metric despite changing the parameters for algorithm then move to step2 and include new variables from the integrated data source.

6. Development of visualizations to communicate results. Thereafter proceed towards step1 and follow the same sequence of steps.

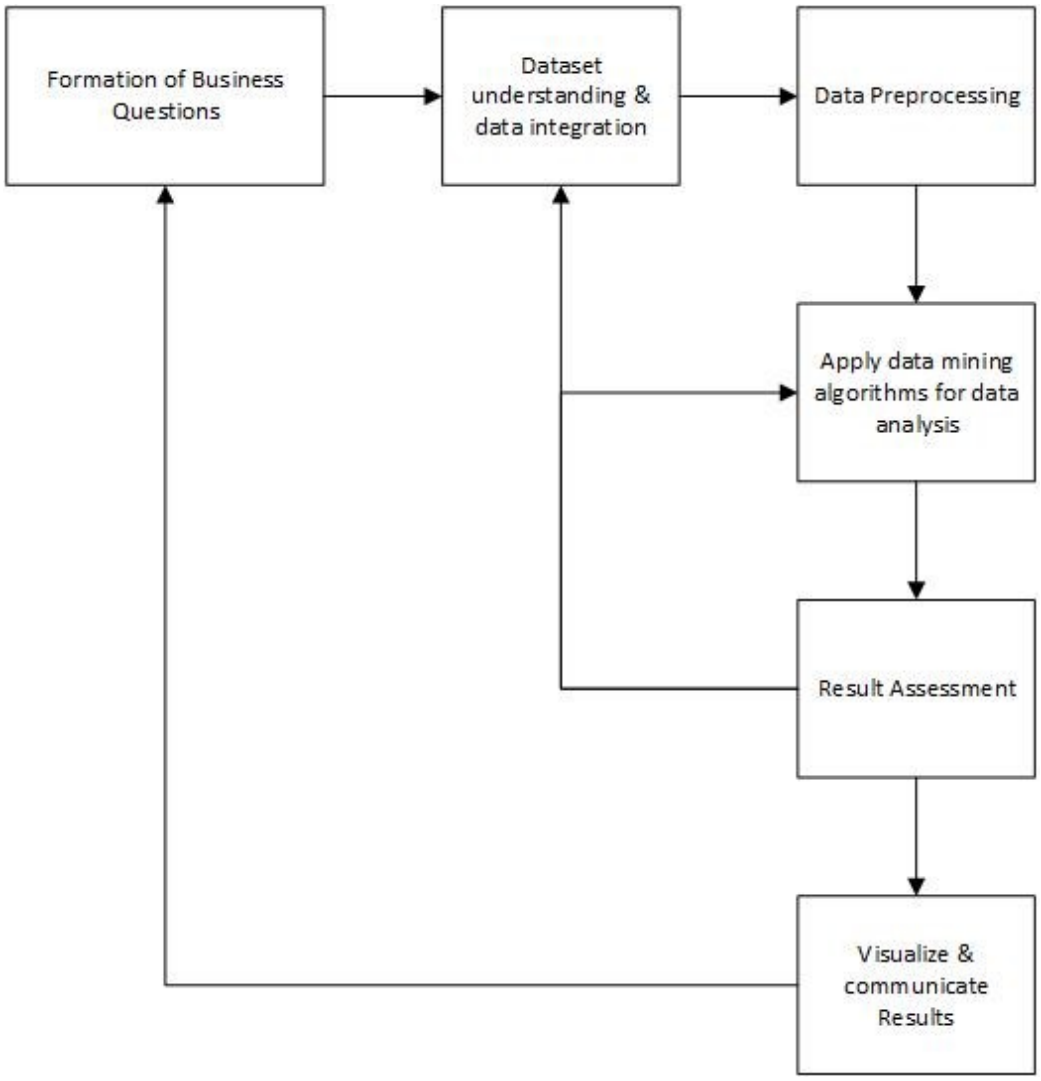

Figure 2.4: Practical Methodology For The Thesis

Chapter 3

Tools & Data Processing

3.1 Tool Selection

The tools are the means to fulfil desired objectives, therefore it is extremely important to make an appropriate selection. The tool selection is restricted by exclusion of licensed products from market leaders like SAS. The tools under consideration for the thesis were as follows:

- **Data Preprocessing**: R
- **Data analysis tools**: R and Weka
- **Data visualization tools**: Tableau
- **Database**: Oracle
- **Programming languages**: Java, JavaScript

R is utilized as a primary data processing tool. *R* has a data structure called *DataFrame* which is similar to a table in a database. *R* is only utilized to process data and to run SVM algorithms due to memory and processing power limitation. *Apache Spark has an advanced DAG execution engine that supports cyclic data flow and in-memory computing.*[33] *Apache Spark* and *Apache SparkMLlib* provide distributable machine learning algorithms for analysis. Apache SparkMLlib has a package Spark.ml which is built on top of *DataFrames*. The familiarity with *DataFrame* which is similar to that of *R*, thus, reduces the learning curve. Apache SparkMLlib does not include algorithms for association rule analysis. *Weka*[22] is used for association rule analysis. *Weka* has a user-friendly interface and it provides implementations of popular machine learning algorithms. *Python* with *Scikit learn*[34] and *Matplotlib*[23] is used to implement unsupervised machine

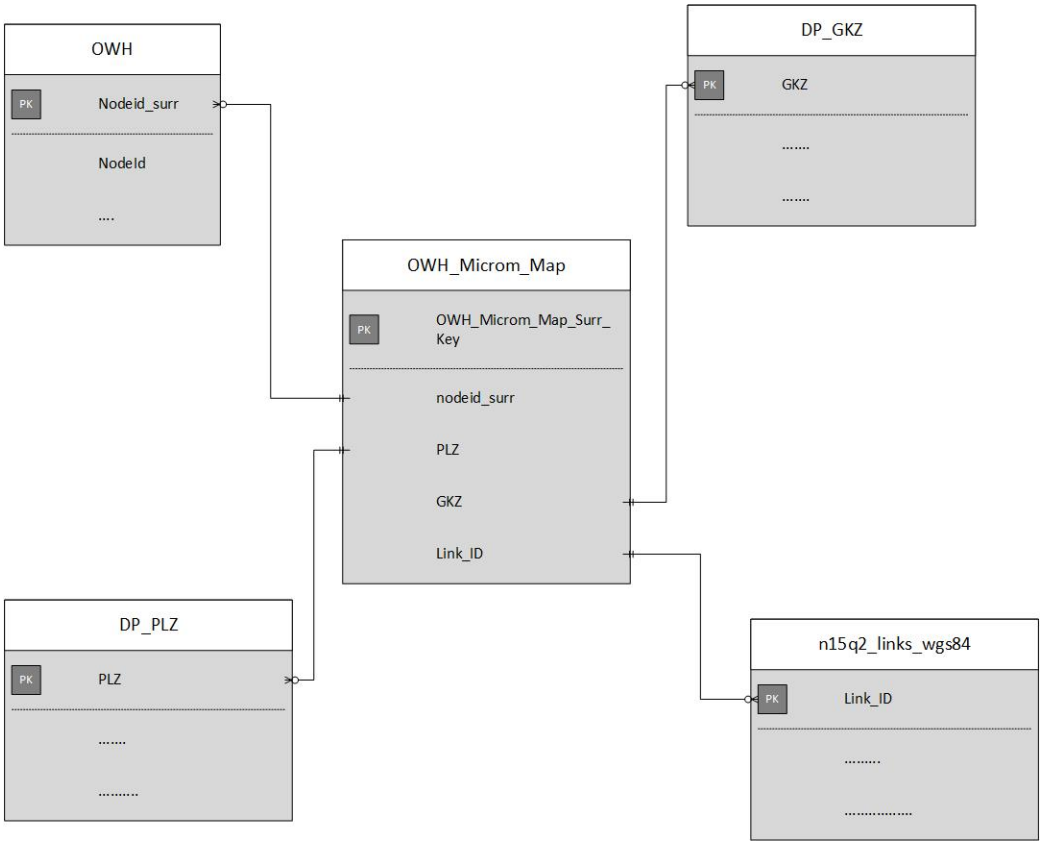

Figure 3.1: Data Model For The Functional Data Warehouse

learning algorithms and produce visualizations. *Tableau* is a business intelligence visualization tool. It is utilized to create visualizations. It provides a simple and user-friendly interface to build charts for analysis.

3.2 Data Integration

The value of integration of the two data sources has been discussed under section 1.3.3. The nature of OWH data source is that of a node dimension. Therefore, MICROM data source and its tables need to be modelled as a dimension for analysis. *Kimball and Ross*[27] in their book suggest the processes to do so.

Figure 3.1 shows the conceptualized data warehouse. As the dimension OWH contains historical data and repetitive nodeids are present therefore it requires a new primary key, also called surrogate keys. The rest of the dimensions are not historical in nature, therefore they need no such mechanism.

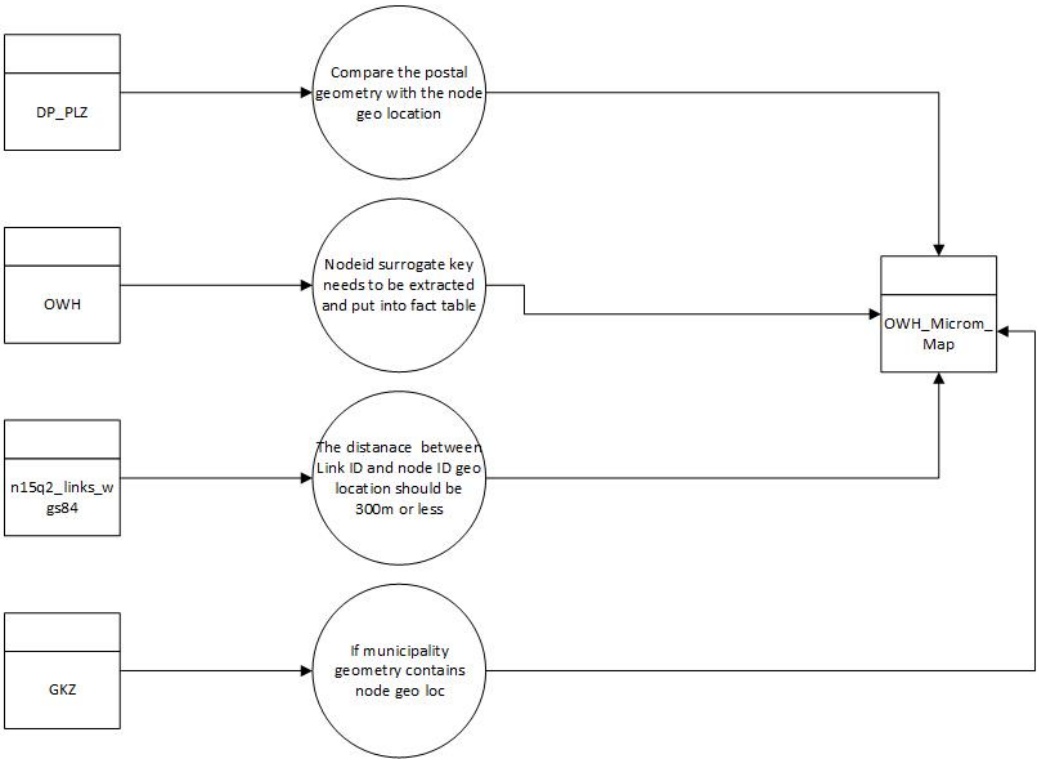

Figure 3.2: ETL For The Data Population

A suggested approach would be to integrate the complete *MICROM* data source with the model. Due to this compromise, only a functional model has been achieved to demonstrate a better method of modelling.

Figure 3.2 shows the ETL steps for populating the data warehouse. The following ETL steps were performed :

- Extract the *NODEID* and generate appropriate Surrogate Keys to populate in the fact table.
- If the *NODEID* from OWH is inside the postal(*PLZ*) geometry then insert the *NODEID* with the postal code inside the mapping table(OWH_Microm_Map)
- If the *NOIDEID* from OWH is inside the Municipal(*GKZ*) geometry then insert the *NODEID* with the municipal code inside the mapping table(OWH_Microm_Map)
- If the distance between *NODEID* from *OWH* and *LinkID* from n15q2_links_wgs84 is 300m or less, then insert *NODEID* and all the *LinkIDs* inside the mapping table(OWH_Microm_Map).

3.3 Variables

Kumar, Vineet and Reinartz, Werner in their book[30] discuss the various methods of variable manipulation - like transformation, elimination and derivation. From the data warehouse, selection of variables for analysis is done by querying the database and then exporting the data to a *CSV* format. The *CSV* file is a compatible format with *R*, *Python*, *Weka* and other tools for data processing and analysis. Although R and other tools are capable of querying the database to generate datasets. The performance and licensed plug-ins are also a constraint for use.

3.3.1 Extracted Variables

There are *2,328,145 observations/rows and 67 variables* as shown in the Figure 3.3, describing each *NODEID* with only the last version from the database that were visible. The variables *POSTALCODE* and *NODEID* are numerical/continuous values on extraction. They need to be converted to categorical values for analysis. The important transformation performed on the directly extracted variables :

1. **TIMESTAMP** :
   The variable *TIMESTAMP* by default is of the string data type. It is transformed to a numeric value for analysis. This is done by the code below.

```
lct<-Sys.getlocale("LC_TIME")
Sys.setlocale("LC_TIME","C")

df_svm$data3Class.TIMESTAMP <-
as.POSIXct(df_svm$data3Class.TIMESTAMP , format="%d-%B-%y")

df_svm$data3Class.TIMESTAMP<-
as.numeric(df_svm$data3Class.TIMESTAMP)
```

2. **TOILETS_WHEELCHAIR, SHOP, TOURISM, SPORT, PUBLIC_TRANSPORT, LEISURE, OFFICE, HISTORIC, AERIALWAY, AEROWAY, BUILDING** :
   The *NULL* values in the variables are replaced with '*not tagged*' value as this is useful for association rule analysis.

3.3.2 Derived Variables

1. **PERCENTAGE_OF_MEN, PERCENTAGE_OF_WOMEN** :
   The variables starting with PLZ_EAG_A provide the numeric information about people between certain age group. For example, the variable *PLZ_EAG_A_M30BIS35* contains the number of men between 30 and 35 years of age and *PLZ_EAG_A_W30BIS35* is the variable containing number of women between 30 and 35 years of age. The absolute numbers are not useful to look for comparisons. Therefore the use of ratios makes more sense. The variables related to age and status are converted to the corresponding percentages for efficient analysis. Therefore, each postal code contains corresponding percentages of men and women. The variables with absolute values are eliminated.

2. **PERCENTAGE_OF_LOW_STATUS_HOUSEHOLDS, PERCENTAGE_OF_AVG_STATUS_HOUSEHOLDS, PERCENTAGE_OF_HIGH_STATUS_HOUSEHOLDS** :
   The status of households with High, Low or Average for the postal area in which the *NODEID* is available.For example *PLZ_MSO_A_STATUS_1, PLZ_MSO_A_STATUS_2, PLZ_MSO_A_STATUS_3+ PLZ_MSO_A_STATUS_4* are the variables related to low-status households. These absolute values to the status values are aggregated to create a corresponding derived variable.

3. **knownunknownflag** :
   The categorical variable WHEELCHAIR_VALID is either known i.e. *yes*, *no* or *limited*, or it is *unknown*. Therefore, a new categorical variable is introduced with two values as *1* if it is known and *0* if it is *unknown*.

4. **GEMEINDENAME** :
   The categorical variable *GEMEINDENAME* stores the name of the municipality where the *NODE* is. There is no direct availability of such information as municipality code is not directly related to the postal code. What that means is that a big municipality has multiple postal codes and if municipalities are small then they are clubbed under one postal code. To solve the issue an intersection between the two is taken and then group by the postal code, and then extracting the maximum *GKZ* value from it, which gives us the name of the municipality. The code below demonstrates the creation of temporary table called *Postalcode_City_Mapping* used for the purpose.

```
create table Postalcode_City_Mapping as
```

```
(select a.*, b.gemeindename  from
(
select plz, max(gkz) keep
(dense_rank last order by c, gkz) max_gkz, max(c) max_c
from (select plz, gkz, count(*)
c from owh_microm_map group by plz, gkz)
group by plz
)  a
join dp_municipality b on b.gkz = a.max_gkz);
```

5. **GERMANSTATE** :
   Germany contains 16 federal states, and the variable *GERMANSTATE* is derived from *GKZ*(Municipality code), the first two digits of the *GKZ* code. For example, **14**612000 => **14**=*'Saxony'*. Thus, using the following method we drive the variable *GERMANSTATE*.

6. **NODEIDCOUNT** :
   The variable is computed during the execution of the program internally and contains the *NODEID* count grouped by *POSTALCODE* and it is a numeric value.

Figure 3.4 shows the all the variables for analysis.

```
$ NODEID                        : num  13 100 123548 128166 128232 ...
$ GEMEINDENAME                  : Factor w/ 4988 levels "Aachen, Stadt",..: 1362 3933 1782 3326 1578 2967 2967 2967 2967 2967 ...
$ POSTALCODE                    : int  34233 29465 25856 82140 82194 81245 81245 81245 81245 81245 ...
$ LON                           : num  9.51 10.83 9.03 11.32 11.39 ...
$ LAT                           : num  51.4 52.9 54.5 48.2 48.2 ...
$ WHEELCHAIR_VALID              : Factor w/ 4 levels "","limited","no",..: 1 1 1 1 4 1 1 1 1 1 ...
$ TOILETS_WHEELCHAIR            : Factor w/ 8 levels "","incorrect",..: 8 1 1 1 1 1 1 1 1 1 ...
$ SHOP                          : Factor w/ 906 levels "","--","*","90969868",..: 1 1 1 1 1 1 1 1 1 1 ...
$ TOURISM                       : Factor w/ 173 levels "","-","agricultural",..: 1 1 1 1 1 1 1 1 1 1 ...
$ SPORT                         : Factor w/ 1095 levels "","*","10pin",..: 1 1 1 1 1 1 1 1 1 1 ...
$ PUBLIC_TRANSPORT              : Factor w/ 38 levels "","abandoned",..: 1 1 1 1 1 1 1 1 1 1 ...
$ AMENITY                       : Factor w/ 849 levels "","-","*","abandoned",..: 1 1 1 1 1 1 1 1 1 1 ...
$ LEISURE                       : Factor w/ 312 levels "","*","04-Bad (Freibad)",..: 1 1 1 1 1 1 1 1 1 1 ...
$ OFFICE                        : Factor w/ 178 levels "","accociation",..: 1 1 1 1 1 1 1 1 1 1 ...
$ HISTORIC                      : Factor w/ 162 levels "","adit","aircraft",..: 1 74 1 1 1 1 1 1 1 1 ...
$ AEROWAY                       : Factor w/ 22 levels "","aerodrome",..: 1 1 1 1 1 1 1 1 1 1 ...
$ AERIALWAY                     : Factor w/ 12 levels "","cable_car",..: 1 1 1 1 1 1 1 1 1 1 ...
$ BUILDING                      : Factor w/ 384 levels "","14","19","4",..: 1 1 1 1 1 1 1 1 1 1 ...
$ USERNAME                      : Factor w/ 48058 levels "","+++++","0025",..: 46343 31980 15174 42680 21833 12552 2048 12552 2048 2048
$ TIMESTAMP                     : Factor w/ 3306 levels "01-APR-07","01-APR-08",..: 1717 89 372 3291 2249 2897 511 2897 2719 2719 ...
$ PURCHASINGPOWERPERHOUSEHOLD: int  45240 36200 44527 55097 59720 59099 59099 59099 59099 59099 ...
$ NUMBEROFHOUSEHOLDS            : int  5650 651 1485 12728 9810 13584 13584 13584 13584 13584 ...
$ NUMBEROFCOMMERCIALBUILDINGS: int  48 13 22 75 25 38 38 38 38 38 ...
$ PLZ_MSO_A_STATUS_1            : int  101 50 15 0 0 34 34 34 34 34 ...
$ PLZ_MSO_A_STATUS_2            : int  298 189 40 159 0 17 17 17 17 17 ...
$ PLZ_MSO_A_STATUS_3            : int  864 168 290 630 0 75 75 75 75 75 ...
$ PLZ_MSO_A_STATUS_4            : int  917 73 343 303 0 328 328 328 328 328 ...
$ PLZ_MSO_A_STATUS_5            : int  934 111 440 399 21 296 296 296 296 296 ...
$ PLZ_MSO_A_STATUS_6            : int  1038 32 189 629 27 769 769 769 769 769 ...
$ PLZ_MSO_A_STATUS_7            : int  833 14 101 2069 222 732 732 732 732 732 ...
$ PLZ_MSO_A_STATUS_8            : int  535 0 50 4534 1850 2113 2113 2113 2113 2113 ...
$ PLZ_MSO_A_STATUS_9            : int  130 14 17 4005 7690 9220 9220 9220 9220 9220 ...
$ PLZ_EWA_A_GESAMT              : int  11942 1354 3297 25486 19270 25793 25793 25793 25793 25793 ...
$ PLZ_EAG_A_M00BIS03            : int  123 10 39 373 247 391 391 391 391 391 ...
$ PLZ_EAG_A_M03BIS06            : int  124 15 46 397 254 363 363 363 363 363 ...
$ PLZ_EAG_A_M06BIS10            : int  174 18 89 503 363 468 468 468 468 468 ...
$ PLZ_EAG_A_M10BIS15            : int  309 36 97 648 516 548 548 548 548 548 ...
$ PLZ_EAG_A_M15BIS18            : int  216 31 68 375 322 311 311 311 311 311 ...
$ PLZ_EAG_A_M18BIS20            : int  153 9 39 251 211 236 236 236 236 236 ...
$ PLZ_EAG_A_M20BIS25            : int  309 31 97 662 459 724 724 724 724 724 ...
$ PLZ_EAG_A_M25BIS30            : int  296 32 71 681 424 970 970 970 970 970 ...
$ PLZ_EAG_A_M30BIS35            : int  269 34 80 832 446 1070 1070 1070 1070 1070 ...
$ PLZ_EAG_A_M35BIS40            : int  293 40 91 839 552 1008 1008 1008 1008 1008 ...
$ PLZ_EAG_A_M40BIS45            : int  429 35 133 1057 705 1103 1103 1103 1103 1103 ...
$ PLZ_EAG_A_M45BIS50            : int  544 54 136 1095 823 1099 1099 1099 1099 1099 ...
$ PLZ_EAG_A_M50BIS55            : int  480 65 130 1030 754 868 868 868 868 868 ...
$ PLZ_EAG_A_M55BIS60            : int  387 50 92 806 598 691 691 691 691 691 ...
$ PLZ_EAG_A_M60BIS65            : int  392 51 108 699 538 664 664 664 664 664 ...
$ PLZ_EAG_A_M65BIS75            : int  794 85 215 1244 1241 1195 1195 1195 1195 1195 ...
$ PLZ_EAG_A_M75UNDGR            : int  613 66 109 821 841 832 832 832 832 832 ...
$ PLZ_EAG_A_W00BIS03            : int  96 15 51 371 220 363 363 363 363 363 ...
$ PLZ_EAG_A_W03BIS06            : int  114 10 43 383 274 341 341 341 341 341 ...
$ PLZ_EAG_A_W06BIS10            : int  161 21 59 419 350 434 434 434 434 434 ...
$ PLZ_EAG_A_W10BIS15            : int  261 49 100 581 422 510 510 510 510 510 ...
$ PLZ_EAG_A_W15BIS18            : int  153 24 63 330 248 294 294 294 294 294 ...
$ PLZ_EAG_A_W18BIS20            : int  99 11 39 264 195 227 227 227 227 227 ...
$ PLZ_EAG_A_W20BIS25            : int  280 22 61 706 428 778 778 778 778 778 ...
$ PLZ_EAG_A_W25BIS30            : int  243 18 73 741 438 1068 1068 1068 1068 1068 ...
$ PLZ_EAG_A_W30BIS35            : int  270 32 86 892 555 1087 1087 1087 1087 1087 ...
$ PLZ_EAG_A_W35BIS40            : int  329 27 94 889 557 937 937 937 937 937 ...
$ PLZ_EAG_A_W40BIS45            : int  418 46 125 1044 768 1015 1015 1015 1015 1015 ...
$ PLZ_EAG_A_W45BIS50            : int  514 51 169 1231 901 1050 1050 1050 1050 1050 ...
$ PLZ_EAG_A_W50BIS55            : int  492 59 120 1010 775 880 880 880 880 880 ...
$ PLZ_EAG_A_W55BIS60            : int  428 40 108 839 582 770 770 770 770 770 ...
$ PLZ_EAG_A_W60BIS65            : int  426 41 99 812 654 762 762 762 762 762 ...
$ PLZ_EAG_A_W65BIS75            : int  873 106 219 1461 1496 1392 1392 1392 1392 1392 ...
$ PLZ_EAG_A_W75UNDGR            : int  880 120 148 1200 1113 1344 1344 1344 1344 1344 ...
```

Figure 3.3: Selected Variables For Analysis

```
$ GEMEINDENAME                    : Factor w/ 4988 levels "Aachen, Stadt",..: 1362 3933 1782 3326 1578 2967 2967 2967 2967 2967 ...
$ NODEID                          : num  13 100 123548 128166 128232 ...
$ POSTALCODE                      : int  34233 29465 25856 82140 82194 81245 81245 81245 81245 81245 ...
$ LON                             : num  9.51 10.83 9.03 11.32 11.39 ...
$ LAT                             : num  51.4 52.9 54.5 48.2 48.2 ...
$ WHEELCHAIR_VALID                : Factor w/ 5 levels "","limited","no",..: 5 5 5 5 4 5 5 5 5 5 ...
$ TOILETS_WHEELCHAIR              : Factor w/ 9 levels "","incorrect",..: 8 9 9 9 9 9 9 9 9 9 ...
$ SHOP                            : Factor w/ 907 levels "","--","*","90969868",..: 907 907 907 907 907 907 907 907 907 907 ...
$ TOURISM                         : Factor w/ 174 levels "","-","agricultural",..: 174 174 174 174 174 174 174 174 174 174 ...
$ SPORT                           : Factor w/ 1096 levels "","*","10pin",..: 1096 1096 1096 1096 1096 1096 1096 1096 1096 1096 ...
$ PUBLIC_TRANSPORT                : Factor w/ 39 levels "","abandoned",..: 39 39 39 39 39 39 39 39 39 39 ...
$ AMENITY                         : Factor w/ 850 levels "","-","*","abandoned",..: 850 850 850 850 850 850 850 850 850 850 ...
$ LEISURE                         : Factor w/ 313 levels "","*","04-Bad (Freibad)",..: 313 313 313 313 313 313 313 313 313 313 ...
$ OFFICE                          : Factor w/ 179 levels "","accociation",..: 179 179 179 179 179 179 179 179 179 179 ...
$ HISTORIC                        : Factor w/ 163 levels "","adit","aircraft",..: 163 74 163 163 163 163 163 163 163 163 ...
$ AEROWAY                         : Factor w/ 23 levels "","aerodrome",..: 23 23 23 23 23 23 23 23 23 23 ...
$ AERIALWAY                       : Factor w/ 13 levels "","cable_car",..: 13 13 13 13 13 13 13 13 13 13 ...
$ BUILDING                        : Factor w/ 385 levels "","14","19","4",..: 385 385 385 385 385 385 385 385 385 385 ...
$ USERNAME                        : Factor w/ 48058 levels "","+++++","0025",..: 46343 31980 15174 42680 21833 12552 2048 12552 2048 2048
$ TIMESTAMP                       : Factor w/ 3306 levels "01-APR-07","01-APR-08",..: 1717 89 372 3291 2249 2897 511 2897 2719 2719 ...
$ PURCHASINGPOWERPERHOUSEHOLD     : int  45240 36200 44527 55097 59720 59099 59099 59099 59099 59099 ...
$ NUMBEROFHOUSEHOLDS              : int  5650 651 1485 12728 9810 13584 13584 13584 13584 13584 ...
$ NUMBEROFCOMMERCIALBUILDINGS     : int  48 13 22 75 25 38 38 38 38 38 ...
$ PLZ_EWA_A_GESAMT                : int  11942 1354 3297 25486 19270 25793 25793 25793 25793 25793 ...
$ PERCENTAGE_OF_MEN               : num  44.3 44 46.4 45.1 43.9 ...
$ PERCENTAGE_OF_WOMEN             : num  50.6 51.1 50.3 51.7 51.8 ...
$ PERCENTAGE_OF_HIGH_STATUS_HOUSEHOLDS: num  11.77 2.15 4.51 67.09 97.25 ...
$ PERCENTAGE_OF_AVG_STATUS_HOUSEHOLDS : num  49.65 24.12 49.16 24.33 2.75 ...
$ PERCENTAGE_OF_LOW_STATUS_HOUSEHOLDS : num  38.58 73.73 46.33 8.58 0 ...
$ knownunknownflag                : num  0 0 0 0 1 0 0 0 0 0 ...
$ GERMANSTATE                     : chr  "Hesse" "Lower Saxony" "Schleswig-Holstein" "Bavaria" ...
```

Figure 3.4: All The Variables Under Analysis

Chapter 4

Algorithm Selection & Theory

There exist a wide variety of algorithms that may be utilized for the thesis. It is not possible to implement and apply all algorithms for analysis, therefore it is important to select a limited number of algorithms that serve our purpose. Research from *Caruana, Rich and Niculescu-Mizil, Alexandru*[6] explores the comparison of algorithms in general. The algorithms listed below have been selected based on popularity, ease of use, efficiency, scalability and previous knowledge.

4.1 Supervised Analysis

A task is called supervised, if the prediction model of data is created, whose target (target/class attribute) is known. The new cases are then derived from the existing data.[36] The most common examples of supervised learning algorithms are decision trees, linear regression, support vector machines,etc. The following questions are evaluated and answered as part of the analysis for classification:

1. Classify if the node is known(value of WHEELCHAIR_VALID variable is either *yes*, *no* or *limited*), or *unknown*.

2. Classify the class of the node as one of the *WHEELCHAIR_VALID* class labels i.e. *yes*, *no* or *limited*.

4.1.1 Decision Trees

Decision trees are widely used since they are easy to interpret, handle categorical features, extend to the multi-class classification setting, do not require

feature scaling, and are able to capture non-linearities and feature interactions. Tree ensemble algorithms such as random forests and boosting are among the top performers for classification and regression tasks.[14]

The decision tree implementation was experimented with R and Weka but due to memory restrictions, the model building process was unsuccessful. As a result, *Spark* with *SparkMLlib* is used to build decision trees. *SparkMLlib* provides data mining algorithms that are distributable. As the data is in the structured format, spark.ml package which is part of SparkMLlib API is built upon *DataFrames*. It is used to train *DecisionTreeClassifier* models and the ensemble models - *RandomForestClassifier* and *GBTClassifier*. The key difference between the ensemble tree models, in statistical language, is that *Random Forests* reduce variance by using more trees, whereas GBTs reduce bias by using more trees. In simple terms, it means that *RandomForestClassifiers* are less prone to over-fitting than *GBTClassifier*.

Decision Tree Classifier

DecisionTreeClassifier is a distributable implementation of decision tree algorithm that is a part of *SparkMLlib* API. The algorithm works on maximizing the *information gain* and the default impurity function used for the calculation is *Gini*. The selected impurity function for the calculation of information gain is *Entropy* because while performing exploratory analysis we intend to maximize mutual information with the tree. For further information on Spark' *DecisionTreeClassifier* algorithm refer to the Apache Spark *DecisionTreeClassifier* documentation[14].

$$Entropy(t) = -\sum p_i log(p_i) \tag{4.1}$$

Notation:
p_i = probability of class i

The list of primary input parameters for the algorithm are :

1. **labelCol**: Param for label column name.
2. **featuresCol**: Param for features column name.
3. **imputury**: Criterion used for information gain calculation (default= Gini).
4. **maxDept**: Maximum depth of the tree (>= 0).

Random Forest

Random forests(Breiman, 2001) is a substantial modification of bagging that builds a large collection of de-correlated trees, and then averages them. Random forests is a notion of the general technique of random decision forests that are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.[18]

Sparkmllib provides a distributable version of the algorithm which helps utilize the cores and the cluster. With the help of distributed computing, the results are obtained efficiently and quickly. For further reference to the '*RandomForest*' algorithm, please visit `http://spark.apache.org/docs/latest/ml-classification-regression.html#random-forests`

The list of primary input parameters for the algorithm are :

1. **labelCol**: Param for label column name
2. **featuresCol**: Param for features column name.
3. **impurity**: Criterion used for information gain calculation (default= Gini).
4. **numTrees**: Number of trees to train (>= 1).
5. **maxDept**: Maximum depth of the tree (>= 0).

Gradient Boosted Trees

Gradient boosting iteratively trains a sequence of decision trees. On each iteration, the algorithm uses the current ensemble to predict the label of each training instance and then compares the prediction with the true label. The dataset is re-labelled to put more emphasis on training instances with poor predictions. Thus, in the next iteration, the decision tree will help correct for previous mistakes.[15] The Sparkmllib implementation for classification is limited to 2-class and the loss function is 'Log Loss' for classification.

$$LogLoss = 2\sum_{i=1}^{N} \log(1 + \exp(-2y_i F(x_i))) \tag{4.2}$$

Notation:
N = number of instances.
y_i = label of $instance_i$.

x_i = features of $instance_i$.
$F(x_i)$ = models predicted label for $instance_i$.

The list of primary input parameters for the algorithm are :

1. **labelCol**: Param for label column name.
2. **featuresCol**: Param for features column name.
3. **imputury**: Criterion used for information gain calculation (default= Gini).
4. **lossType**: Loss function which GBT tries to minimize (default=Log Loss).
5. **maxDept**: Maximum depth of the tree (>= 0).

4.1.2 Logistic Regression

Logistic regression is a mathematical modelling approach that can be used to describe the relationship of several Xs to dichotomous dependent variable[28]. Spark.ml package supports binary logistic regression classification as if now which will be extended to multi-class in the future.[16]

$$L(\mathbf{w}; \mathbf{x}, y) = \log(1 + \exp(-yw^T x)). \tag{4.3}$$

Notation:
x = New data point
L(w;x,y) = logistic loss

List of primary input parameters for the algorithm:

1. **elasticNetParam**: the ElasticNet mixing parameter, in range [0, 1]. For alpha = 0, the penalty is an L2 penalty. For alpha = 1, it is an L1 penalty (default: 0.0)
2. **featuresCol**: features column name (default: features)
3. **maxIter**: maximum number of iterations (>= 0) (default: 100)
4. **regParam**: regularization parameter (>= 0) (default: 0.0)
5. **standardization**: whether to standardize the training features before fitting the model (default: true)

4.1.3 Support Vector Machine

Support vector machines are the most well-known of a class of algorithms that use the idea of kernel substitution and which we will broadly refer to as kernel methods[3]. A Support vector machine model is a representation of the examples as points in space mapped to separate categories divided by a clear gap that is as wide as possible. Thereafter predictions or classifications are made based on where the test examples lie to the SVM decision boundary. R package *e1071*[10] provides an interface to the *libsvm*[7]. The various kernels used for SVM model building are stated below:

$$\textbf{Linear} : u' * v \tag{4.4}$$

$$\textbf{Polynomial} : (gamma * u' * v + coef0)^{degree} \tag{4.5}$$

$$\textbf{Radial basis} : exp(-gamma * |u - v|^2) \tag{4.6}$$

$$\textbf{Sigmoid} : tanh(gamma * u' * v + coef0) \tag{4.7}$$

4.1.4 Neural Network - Multi-layer Perceptron

Multilayer perceptron classifier (MLPC) is a classifier based on the feed-forward artificial neural network. MLPC consists of multiple layers of nodes. Each layer is fully connected to the next layer in the network. Nodes in the input layer represent the input data. [17] Spark.ml package from Apache provides a distributable algorithm to implement multilayer perceptron. The primary inputs to the algorithm are :

1. **layers**: Layer sizes including input size and output size.
2. **featuresCol**: features column name (default: features)
3. **labelCol**: Param for label column name.

4.2 Unsupervised Analysis

Unsupervised learning is a type of machine learning algorithm used to draw inferences from datasets consisting of input data without labelled responses[37]. Unsupervised analysis helps find interesting patterns in the data exploratively without looking for predefined questions. The most known classes of unsupervised learning are clustering and association rules.

4.2.1 Clustering

Cluster analysis is the formal study of methods and algorithms for grouping or clustering objects according to measured or perceived intrinsic characteristics or similarity[25]. As the nature of the analysis is explorative and also including geo-spatial data, therefore clustering is an attractive analytic approach to finding interesting patterns. *Scikit-learn*[34], a python library for machine learning provides a module called *sklearn.cluster* for cluster analysis. Scikit-learn is a well documented machine learning Python API and *sklearn.cluster* has implementations of various clustering algorithms like density based clustering, agglomerative clustering, k-means, etc.

K-means

The KMeans algorithm clusters data by trying to separate samples in n groups of equal variance, minimizing a criterion known as the inertia or within-cluster sum-of-squares.The K-means algorithm aims to choose centroids that minimise the inertia, or within-cluster sum of squared criterion[39]. The algorithm implementation of minimizing sum of squared criterion by the *K-means* algorithm is called *KMeans++*. It improves on the assumption of randomly chosen centroids by the algorithm itself.

The implementation of *K-means* in *Scikit-learn* has the following important input parameters:

1. **n_clusters** : The number of clusters to form as well as the number of centroids to generate.

2. **max_iter** : Maximum number of iterations of the k-means algorithm for a single run.

3. **n_init** : Number of time the k-means algorithm will be run with different centroid seeds. The final results will be the best output of n_init consecutive runs in terms of inertia.

4. **init** : The way to choose initial centroids. Preferable and value is *k-means++* that provides better centroids for analysis.

Agglomerative Clustering

Hierarchical clustering is a general family of clustering algorithms that build nested clusters by merging or splitting them successively[41]. Agglomerative clustering creates clusters by recursively merging the pair of clusters that minimally increases a given linkage distance.

The implementation of *AgglomerativeClustering* in *Scikit-learn* has the following important input parameters:

1. **linkage** :The linkage criterion that determines which distance to use between observations. The default value of the parameter is *ward*, it minimizes the variance of the clusters being merged. The other values are *complete* and *average*.

2. **n_clusters** :The number of clusters to form.

3. **metric**: The metric to use when calculating distance between instances in a feature array, for example euclidean distance.

4. **min_samples** : The minimum number of points required to form a dense region.

Density Based Clustering(DBSCAN)

The DBSCAN algorithm views clusters as areas of high density separated by areas of low density[4]. As the data is clustered based on density,thus, it can take different shapes.

The implementation of *DBSCAN* in *Scikit-learn* has the following input parameters :

1. **eps** :The maximum distance between two samples for them to be considered as in the same neighbourhood.

2. **metric** : The metric to use when calculating distance between instances in a feature array, for example euclidean distance.

3. **min_samples** : The minimum number of points required to form a dense region.

4.2.2 Association Rules - Apriori

Association rule analysis is about finding interesting relations between variables. Apriori[1], a popular Association rule analysis is a well-known algorithm for association rule data mining. The algorithm works on finding the frequent item-sets which have a minimum *support*. For the purpose of implementation, the missing values of the nominal variables are replaced by '*not tagged*' because there is a possibility of an interesting relationship or pattern. *Weka* provides a wide range of algorithms of association rule analysis which are easy to use and analyse.

4.3 Evaluation Metric For Analysis

To compare the algorithms and their performance, there is a requirement of an evaluation metric. The evaluation metric for the supervised analysis is *Accuracy and Error*[29], while testing the model built. The test methodology is using 70% of the data to build the model and 30% of the data for testing. The metric of accuracy and test error are chosen as they provide the measure of correctly classified test samples. Accuracy is the number of observations that are classified correctly divided by the total number of provided observations. And the test error is the subtraction of accuracy from unity/one.

While for unsupervised cluster analysis is evaluated using *Silhouette Coefficient*[2], it is ideal for unsupervised analysis when the class labels are unknown. *Silhouette Coefficient* takes the value from [-1 to 1] and evaluates both inter-cluster and intra-cluster distance. Therefore value closer to 1 indicates that the cluster is well defined while a negative value indicates that the samples have been incorrectly placed to create the clusters.[34]

Chapter 5

Analysis & Results

The nature of the thesis is exploratory, therefore, the questions need to be categorized and answered accordingly. The analysis is divided into three sections that is visual analysis showing graphs to compare and analyse the tagging activity, supervised analysis to answer classification questions and unsupervised analysis to explore and find the interesting collection of features through clustering or other unsupervised methods[20].

5.1 Visual Analytics

Visual analytics is the science of analytical reasoning facilitated by interactive visual interfaces[38]. It helps us explore, describe and analyse data efficiently with the use of graphs and charts. The visual analytics as a part of the thesis provides an extended picture of the tagging activity. The focus of visual analytics as a part of the thesis is also to recommend visualizations to the *CAP4Access community* as a part of their dashboard for future use.

5.1.1 Time-line View Of The Tagging Activity

The data collected as a part of the *CAP4Access*[11] project is collected both through crowd sourcing event campaigns and from motivated individual users that are not a part of the *CAP4Access*[11] project. The time-line helps analyse tagging trend/activity with respect to time. The total number of marked records so far is 482,383 and the number of unknown records is 1,845,732; which is 20.7% of the total number of records, which is shown in the Figure 5.1. This leads us to begin our analysis, the first analysis is to answer what is the time-line history of the tagging activity.

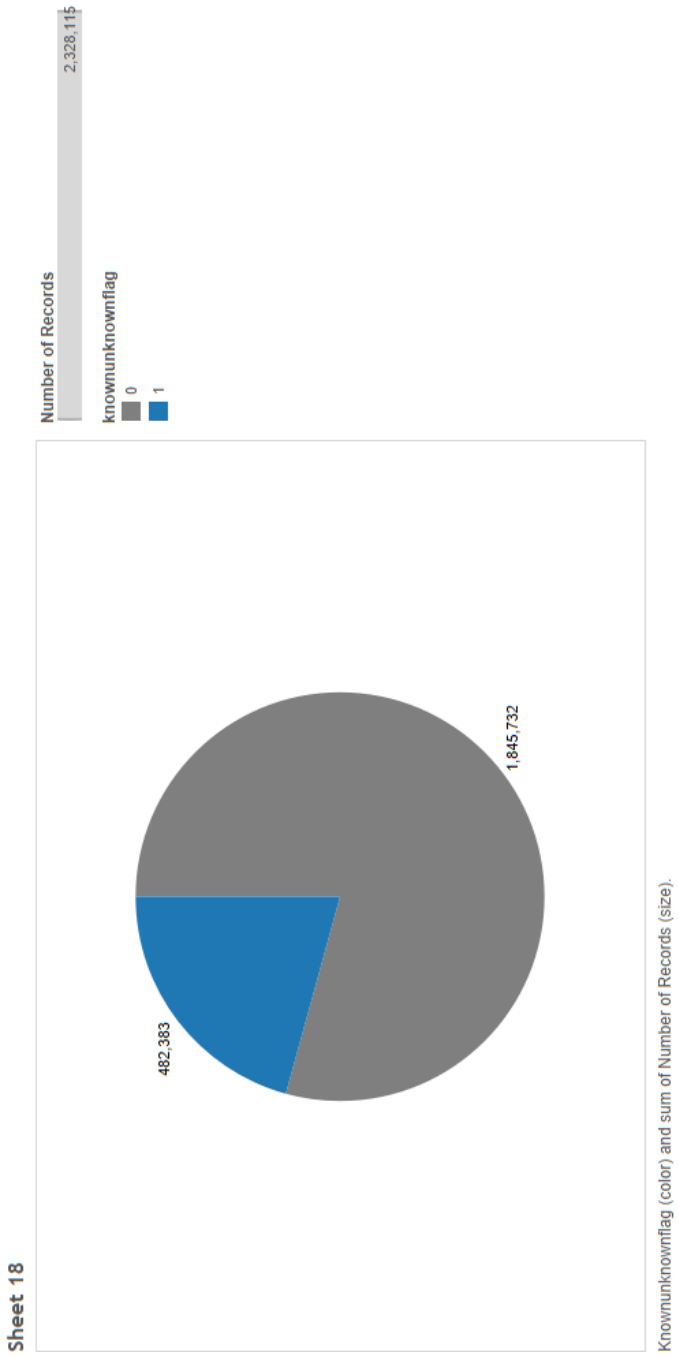

Figure 5.1: Pie Chart To Show The Marked And Unmarked Records

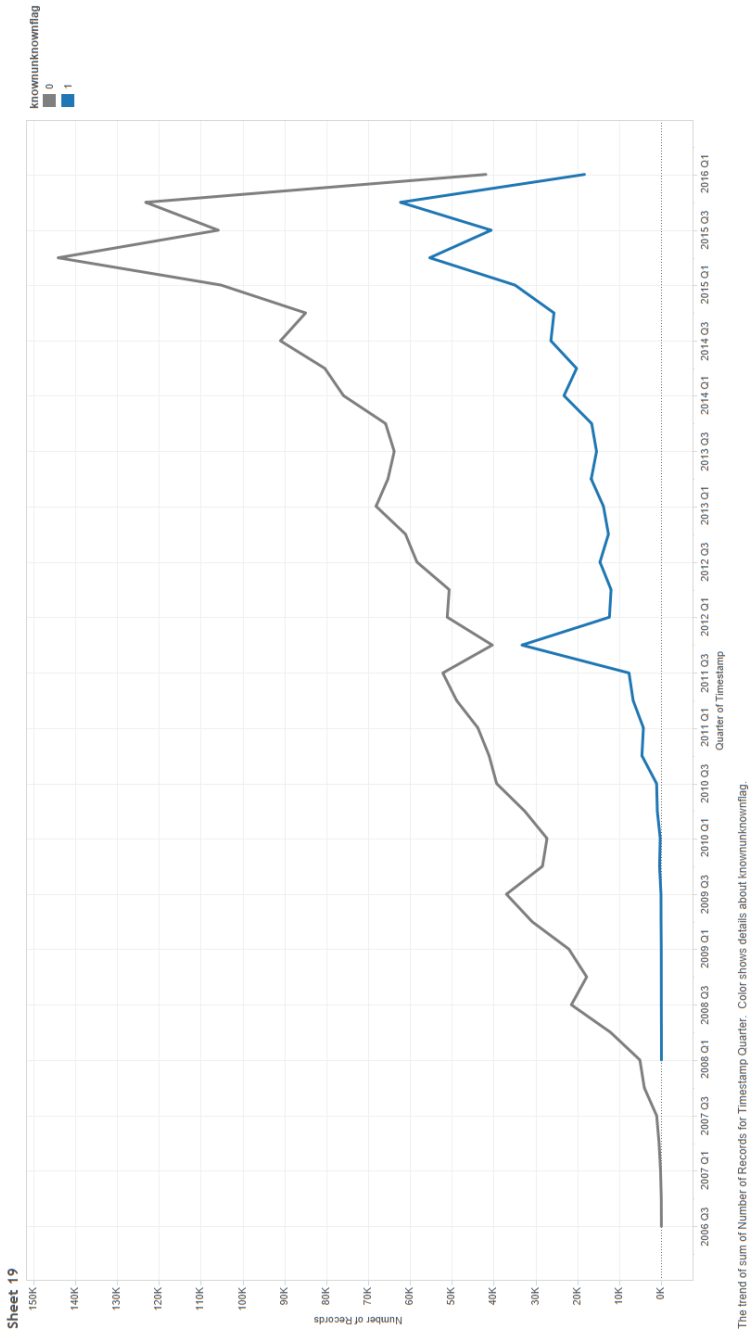

Figure 5.2: Time-line To Show The Number Of Records For The Variable WHEELCHAIR_VALID

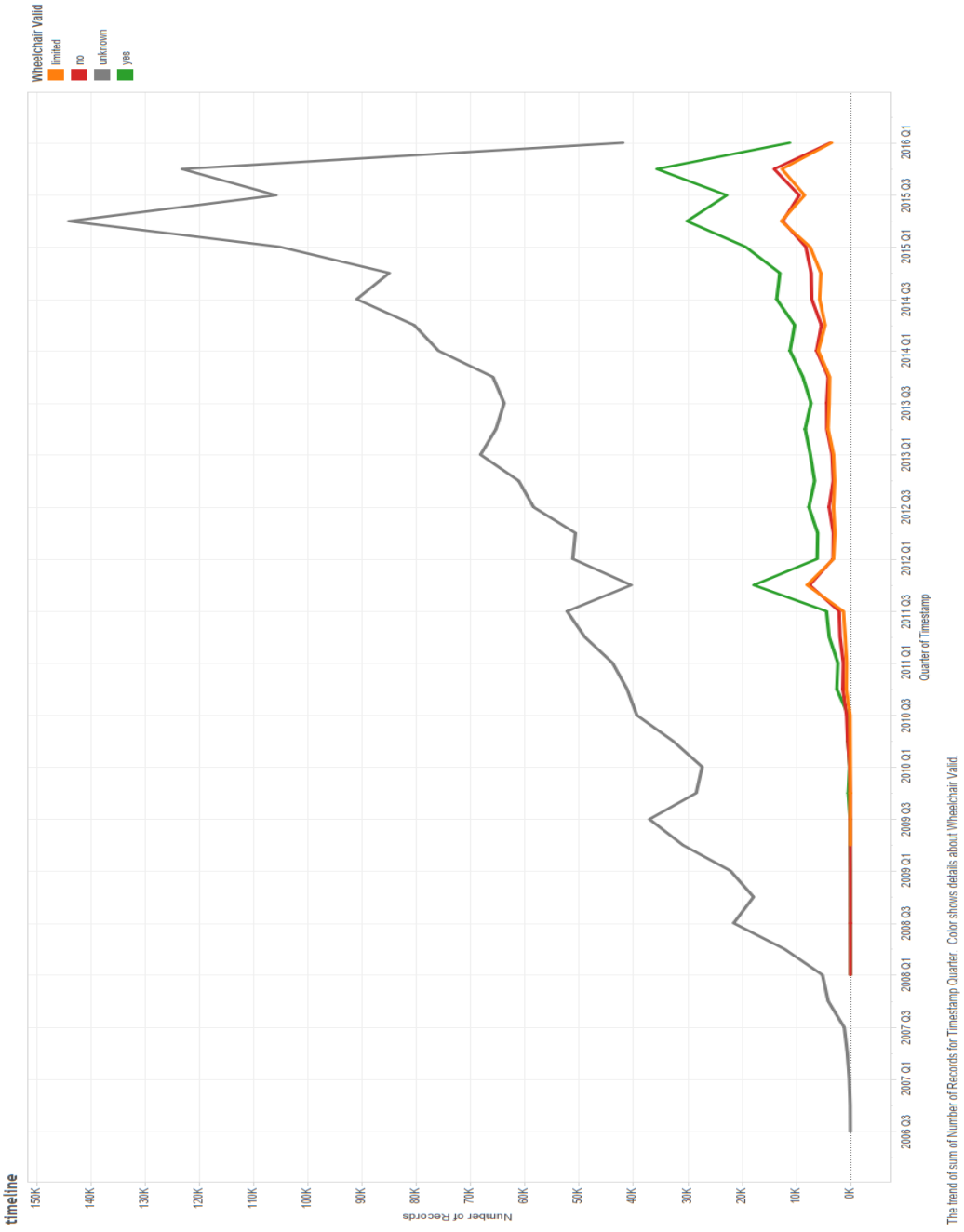

Figure 5.3: Time-line To Show The Number Of Records For The Variable WHEELCHAIR_VALID

The statistics in Figure 5.2 show the time-line marking activity quarterly, this extends the confidence in the data shown by the pie chart i.e. the number of places tagged which is represented by the label *1* during the activity is considerably lesser than the places that are not tagged which is represented by the label *0*. The best tagging activity happens in quarter 4 of the year 2011 where the two lines are the closest together. The tagging activity gets more aggressive with time. Figure 5.3 shows the breakdown of the tagged places into *yes*, *no* and *limited* categories. The time-line showing places tagged as *yes* being more in number considerably than places tagged as *no* and *limited*. The probable reasons for this behaviour are that either the users preferably look for accessible places and when they are in doubt they do not mark the place at all, or the users are visiting popular places which are already accessible. The time-line for categories *no* and *limited* overlap with the exception of a few quarters.

5.1.2 Geographical Analysis Of The Tagging Activity

Germany has 16 states with North Rhine-Westphalia being the most populated, followed by *Bavaria* and *Baden-Wuerttemberg*. The state with the least population state is *Bremen*. The biggest state in Germany by area is Bavaria followed by Lower Saxony and Baden-Wuerttemberg, while Bremen is the smallest state by area[8]. Thus, the next question to analyse is the tagging activity in the German states and cities.

Figure 5.4 displays German states sorted by the sum of records showing the tagging activity for each. The graphic is sorted by the total number of nodes available for tagging. *North Rhine-Westphalia* is at the top of the list, it is the most populated state of Germany and shows the most activity. *Bayern/Bavaria* and *Lower Saxony* which are the biggest states of Germany in terms of area, come at fourth and third positions on the table. But the common behaviour to observe is that the tagged nodes are only a fraction of the nodes available for tagging. Thus, *unknown* nodes are in the extreme majority, this is also a sign that the awareness movements are in an early phase which is accelerating with time.

Figure 5.5 displays the distribution of the tagged nodes on the map in blue. The states of *Bavaria* and *Thuringia* are the least sparsely tagged. Also, the east part of Germany is sparsely tagged compared to the West. Figure 5.6 shows the distribution of the known tags that is *yes*, *no* and *limited* which follows a similar pattern as the distribution in Figure 5.5.

The next analysis question is that which cities of Germany are the most active in the tagging activity, which is shown in Figure 5.7. The four biggest cities of Germany i.e. Berlin, Munich, Hamburg and Cologne are in the top

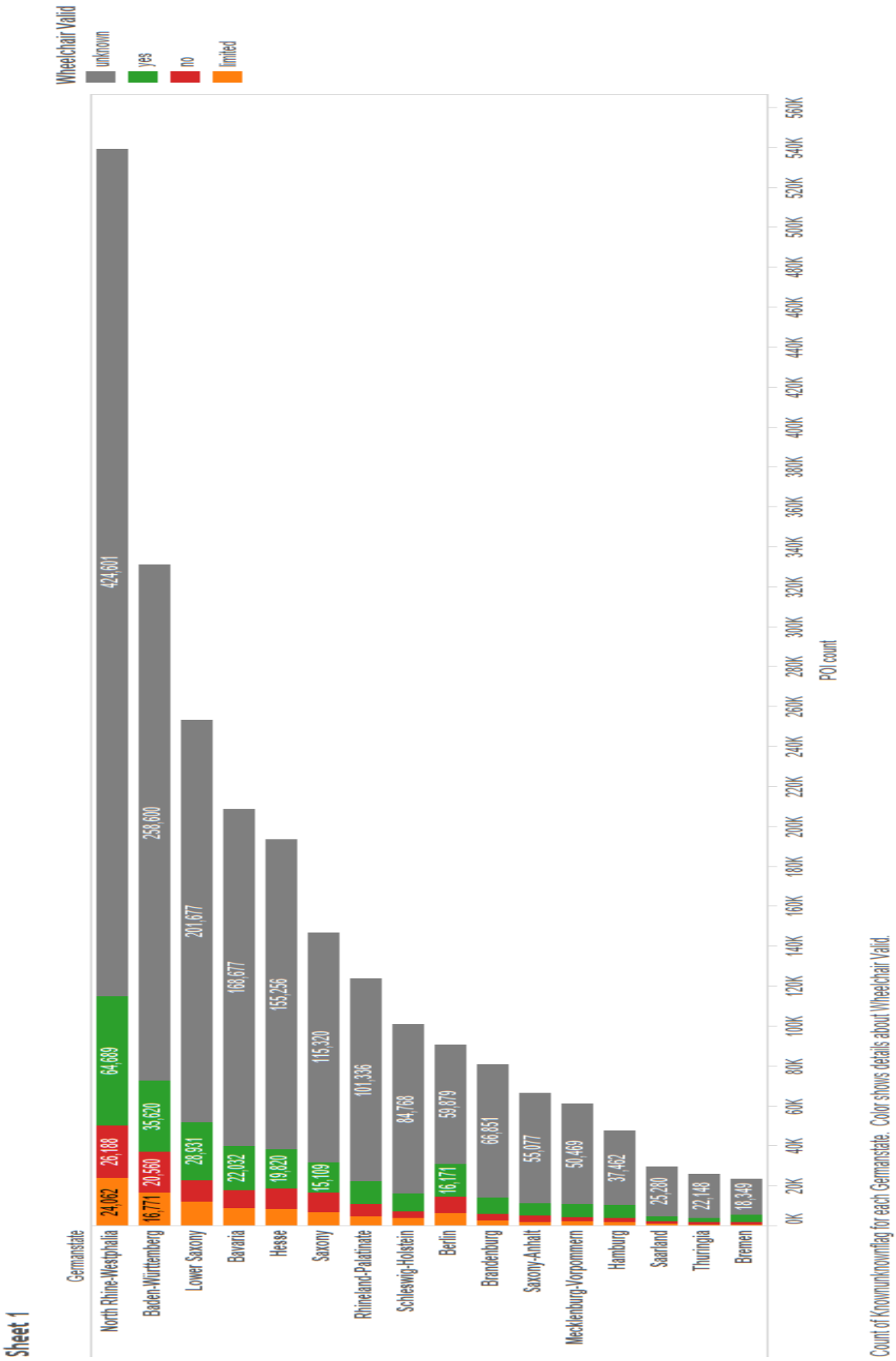

Figure 5.4: State Ranking Based On POI Count In Decreasing Order

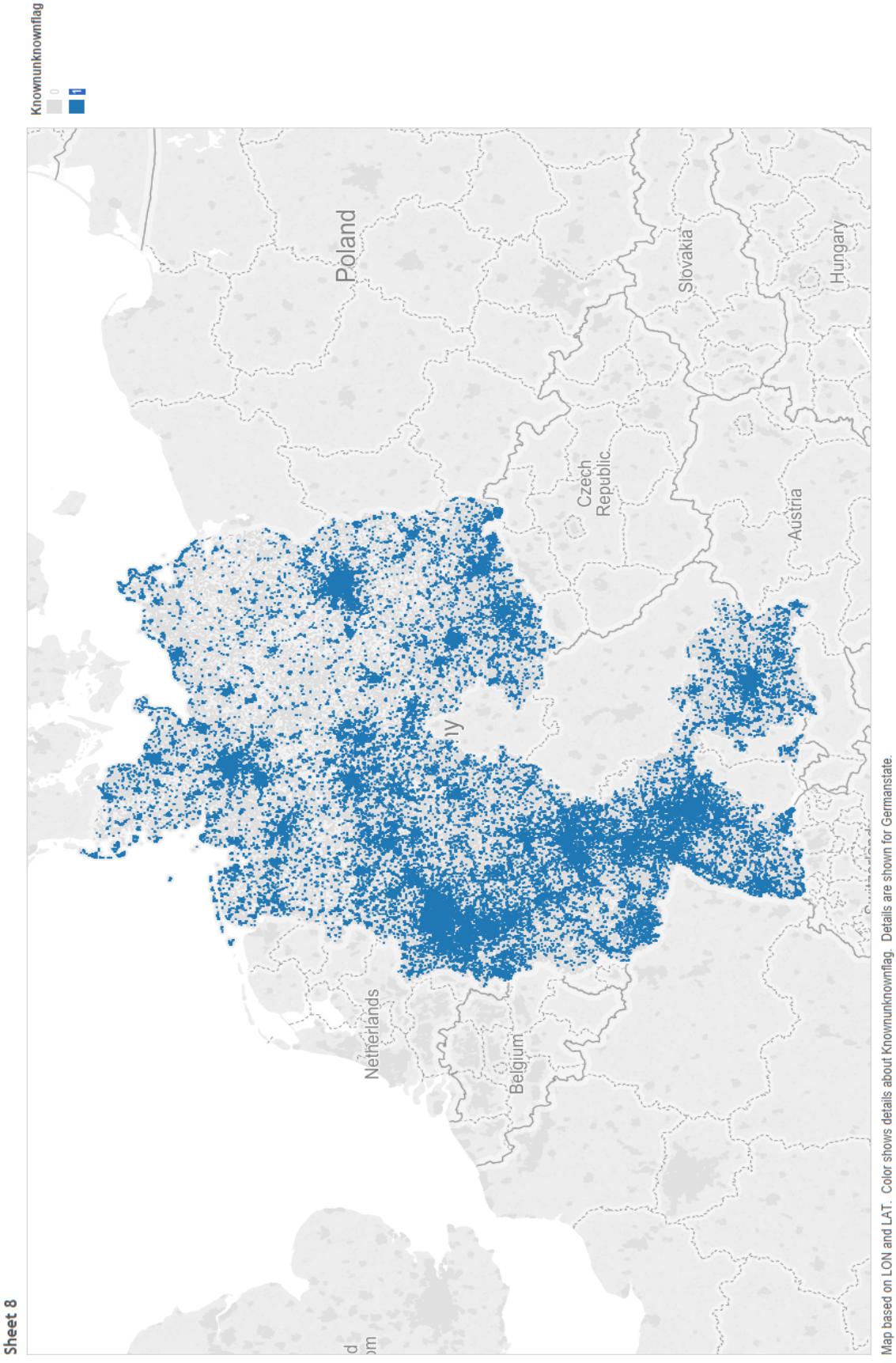

Figure 5.5: Scatter Plot Displaying The Known POIs On The Map

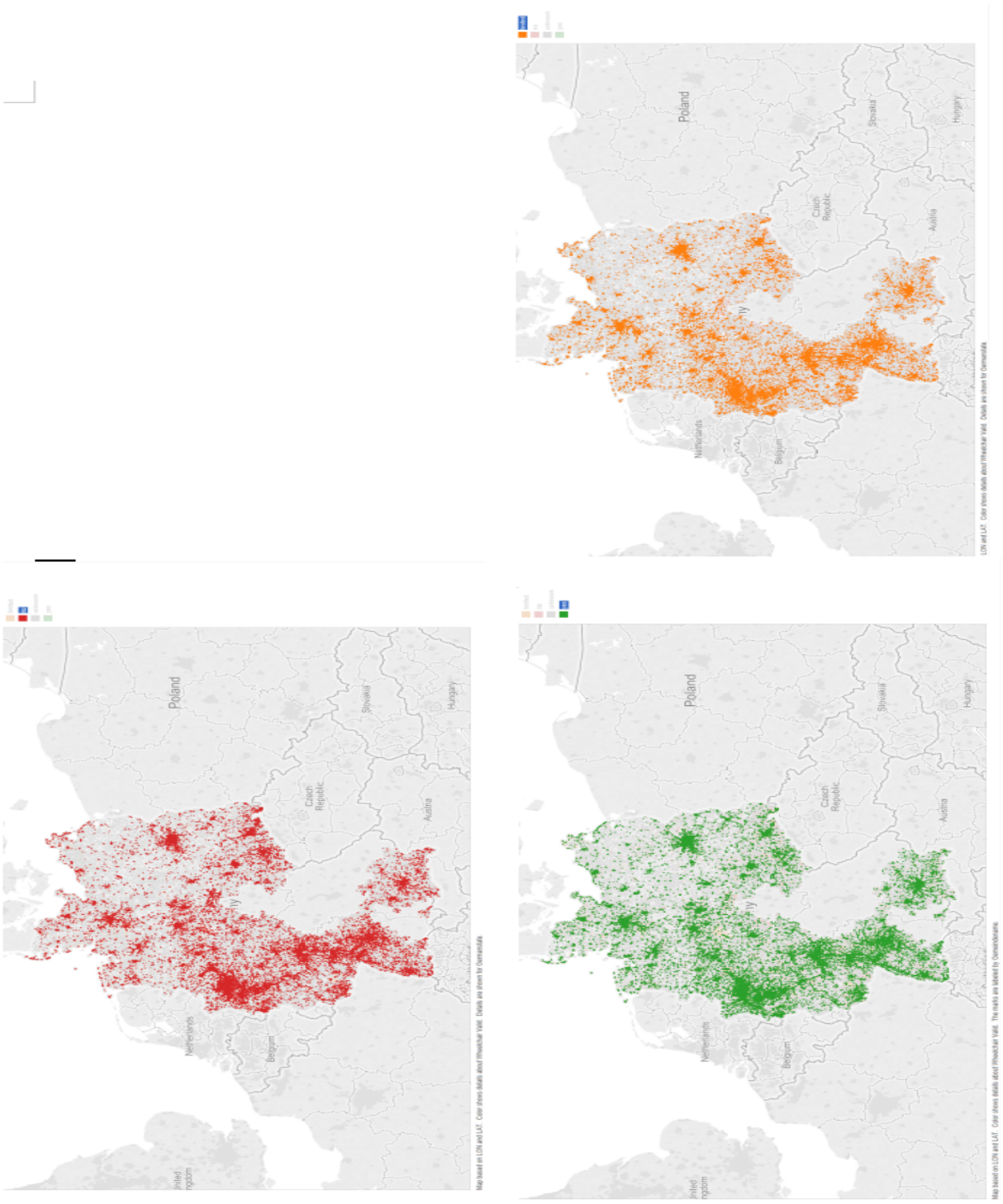

Figure 5.6: Scatter Plot Displaying Individual Known Values that is *no* In Red, *yes* In Green And *limited* In Orange On The Map For Various WHEELCHAIR_VALID Values

half of the list. It is also quite interesting to see that the most populated state of Germany, North Rhine-Westphalia has seven cities in the list while states *Thuringia*, *Saarland*, *Saxony-Anhalt* and *Schleswig-Hostein* have no cities in the top 20. It is also worth noting that Berlin, the state and Berlin, the city share the same data. The pilot site *Heidelberg* does not figure in the top 20, which is quite surprising, this may be because it is a small city.

5.1.3 Analysis Of Marked Variables

Analysis of categorical variables tagged as a part of the *CAP4Access*[11] project provides the most frequently tagged category or place by the user, thus, help us understand the user behaviour. Because there are a large number of categorical values with very low occurrences, therefore, a top twenty list gives a better picture of the

Analysis of variable *AEROWAY* shows that there is extremely small tagging activity with only 4 categories i.e. *terminal*, *aerodrome*, *helipad* and *hangar*. *Terminal* being the mode of the variable *AEROWAY*. Figure 5.8 shows the complete statistic for the same.

The variable *TOURISM* has the first twenty categories which are extremely active these have been shown in Figure 5.9. The mode for the variable *TOURISM* is *hotel*, followed *attraction* and *museum*.

Variable *BUILDING* contains the buildings which were tagged by the users and it has *yes* as the mode which is followed by *church, commercial* and *retail* but what is quite important to see is that places like *hospital* are in the last three. Figure 5.10 shows the statistic for the same.

Variable *HISTORIC* includes categorical values or places of historical importance that were tagged by the users. The mode for the variable *HISTORIC* is *memorial* , followed by *castle*.Figure 5.11 displays the statistic for the variable *Historic*. Historical churches are the third on the list but they were the second on the list of most tagged places for buildings.

Variable *LEISURE* includes the categorical values related to sports or health activities. Figure 5.12 shows the top twenty *LEISURE* places tagged by the users which are accessible. *Playground* is the mode for this variable, followed by *sport centre*, *pitch* and *swimming pool* respectively.

Public transports are the most important interest areas for the people on wheelchairs or people with movement disabilities, the variable *PUBLIC TRANSPORT* includes such tagged places. As seen from Figure 5.13 *platform* is the most tagged value of the variable, followed by *stop position* and *station*.

The variable *SHOPS* is the variable with the one of the highest tagging activity, it has the mode as *supermarket*, followed by *bakery*, *clothes* and

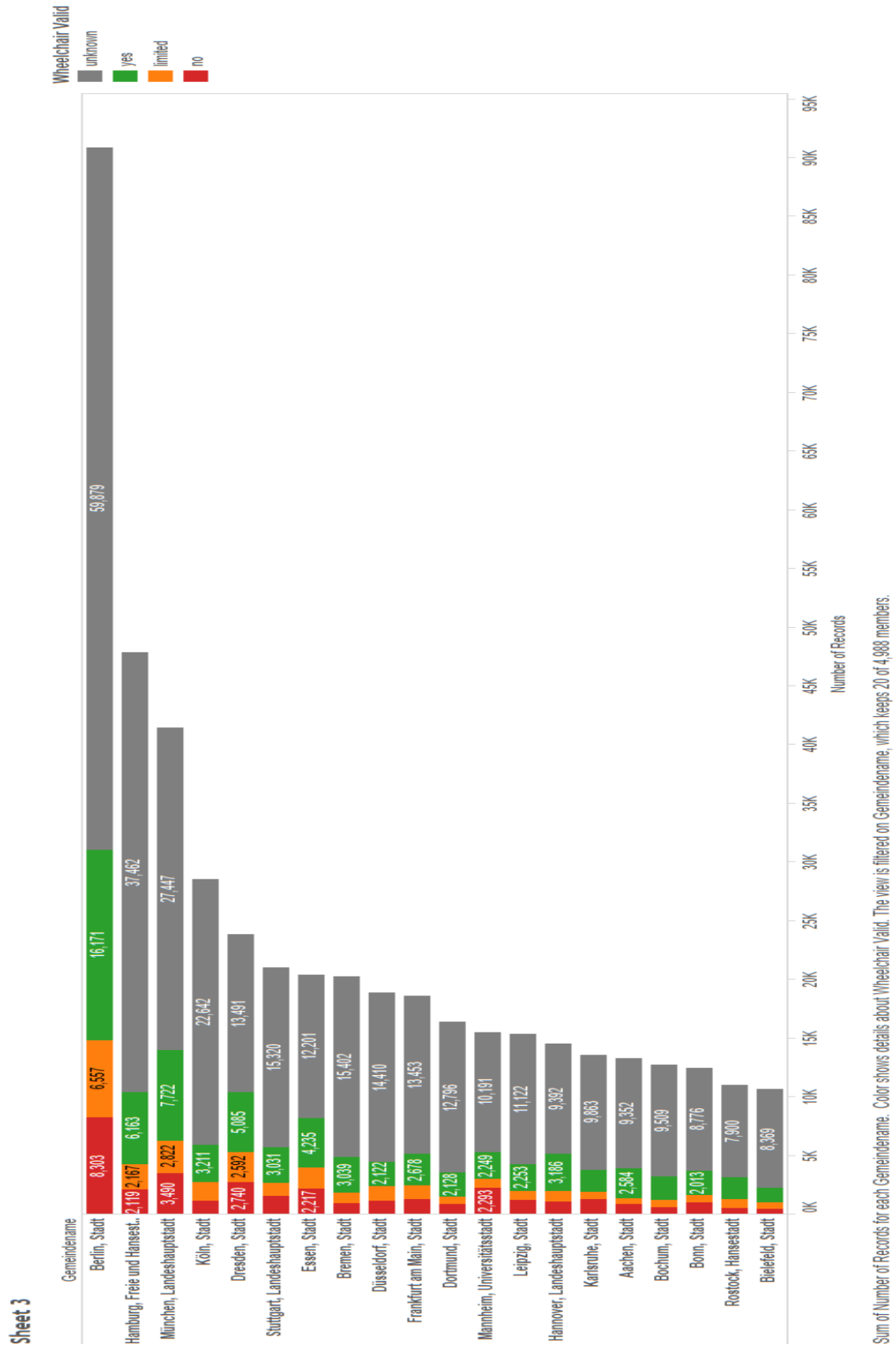

Figure 5.7: Top Cities Based On POI Count In Germany In Decreasing Order

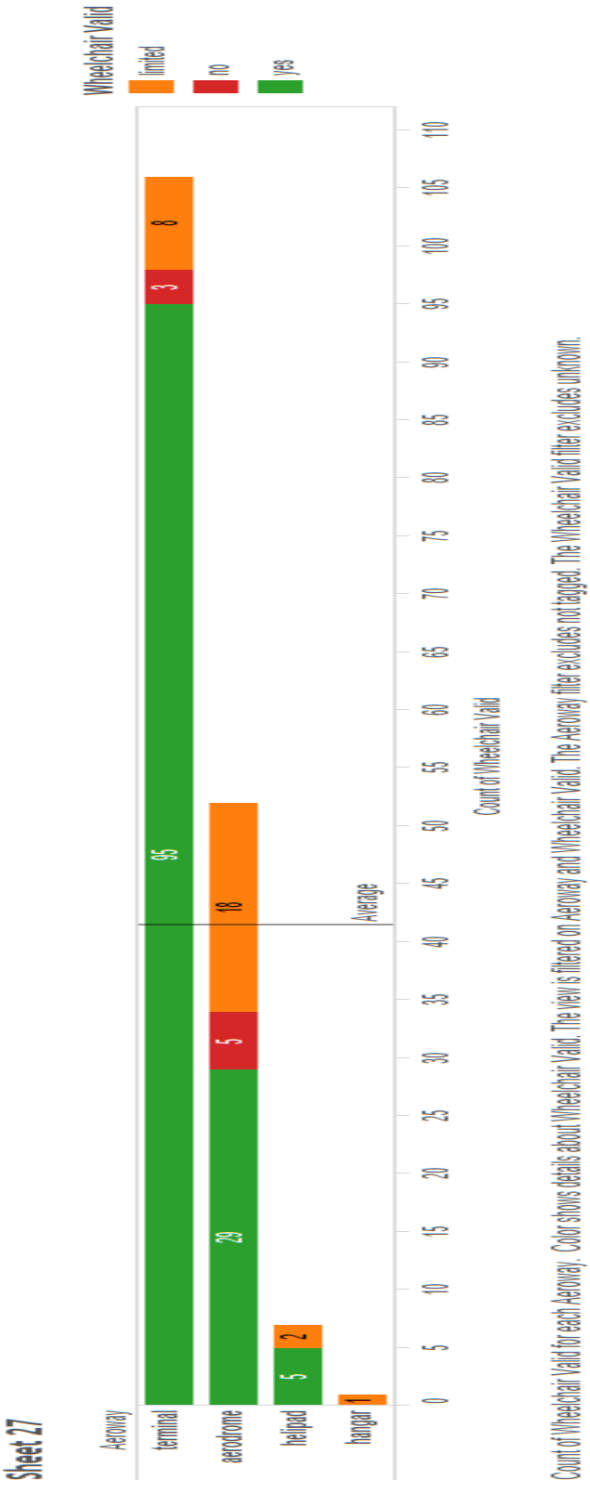

Figure 5.8: Values & Ranking Based On The Total POI count For The Variable *AEROWAY*

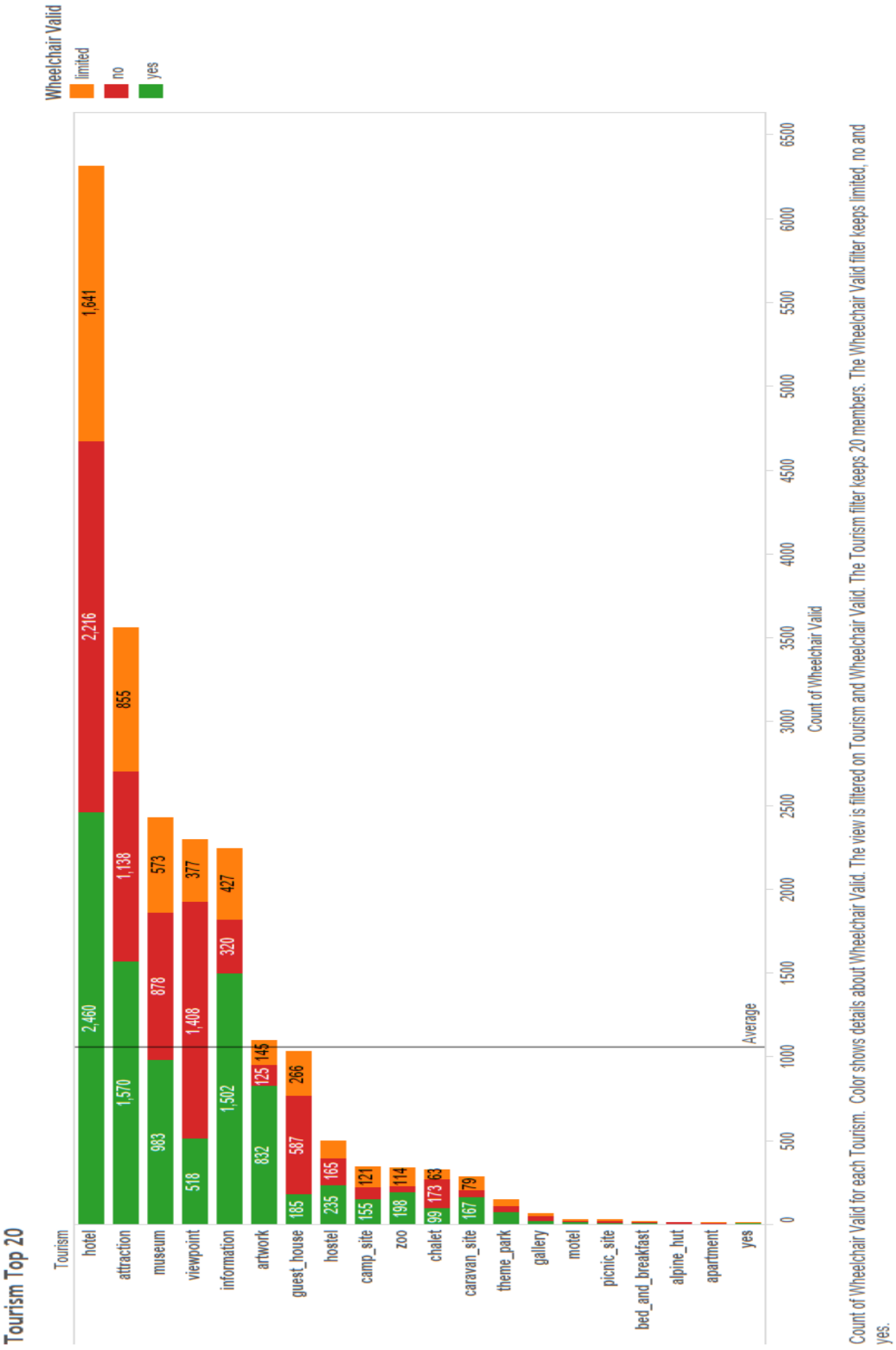

Figure 5.9: Values & Ranking Based On The Total POI count For The Variable *TOURISM*

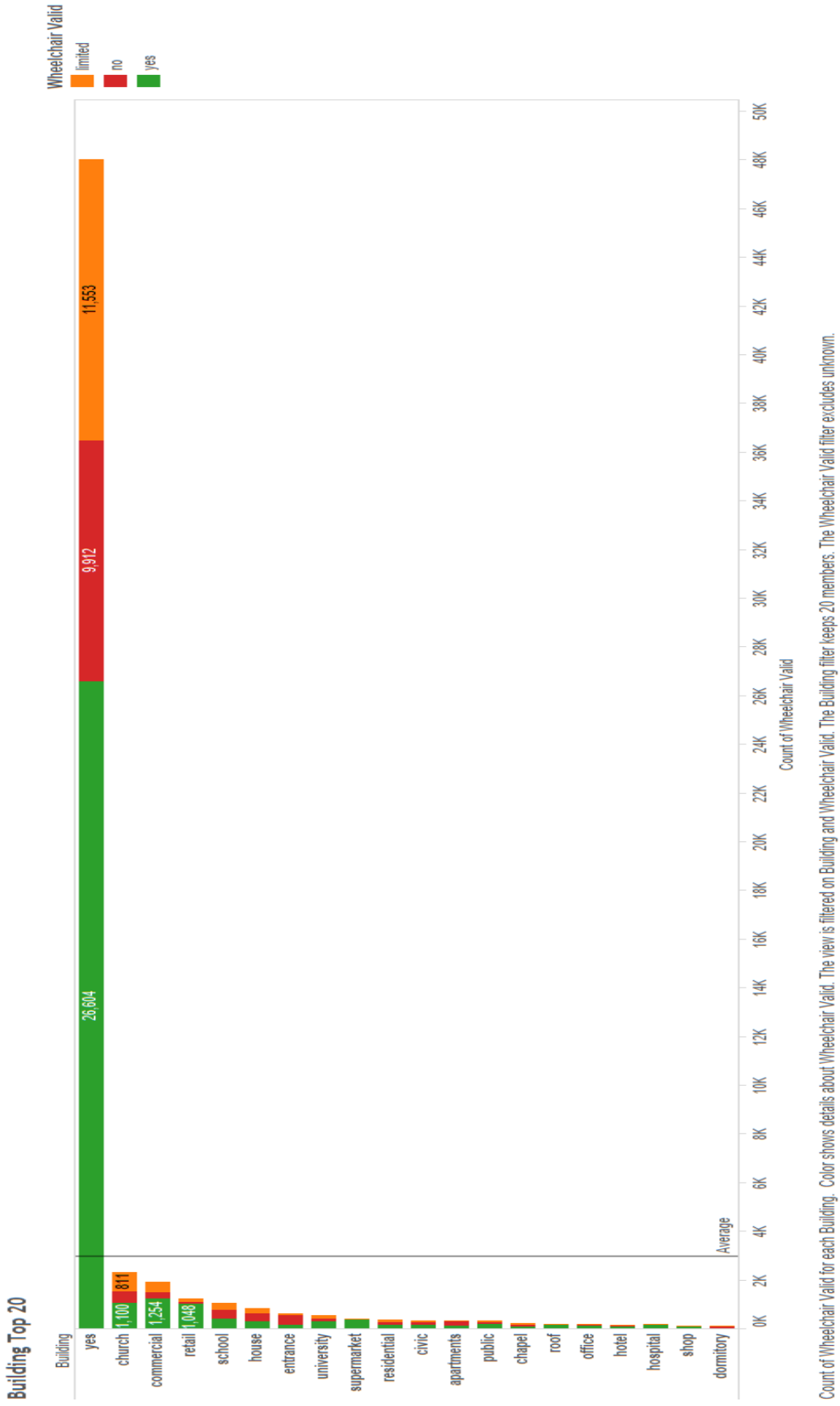

Figure 5.10: Values & Ranking Based On The Total POI count For The Variable *BUILDING*

Figure 5.11: Values & Ranking Based On The Total POI count For The Variable *HISTORIC*

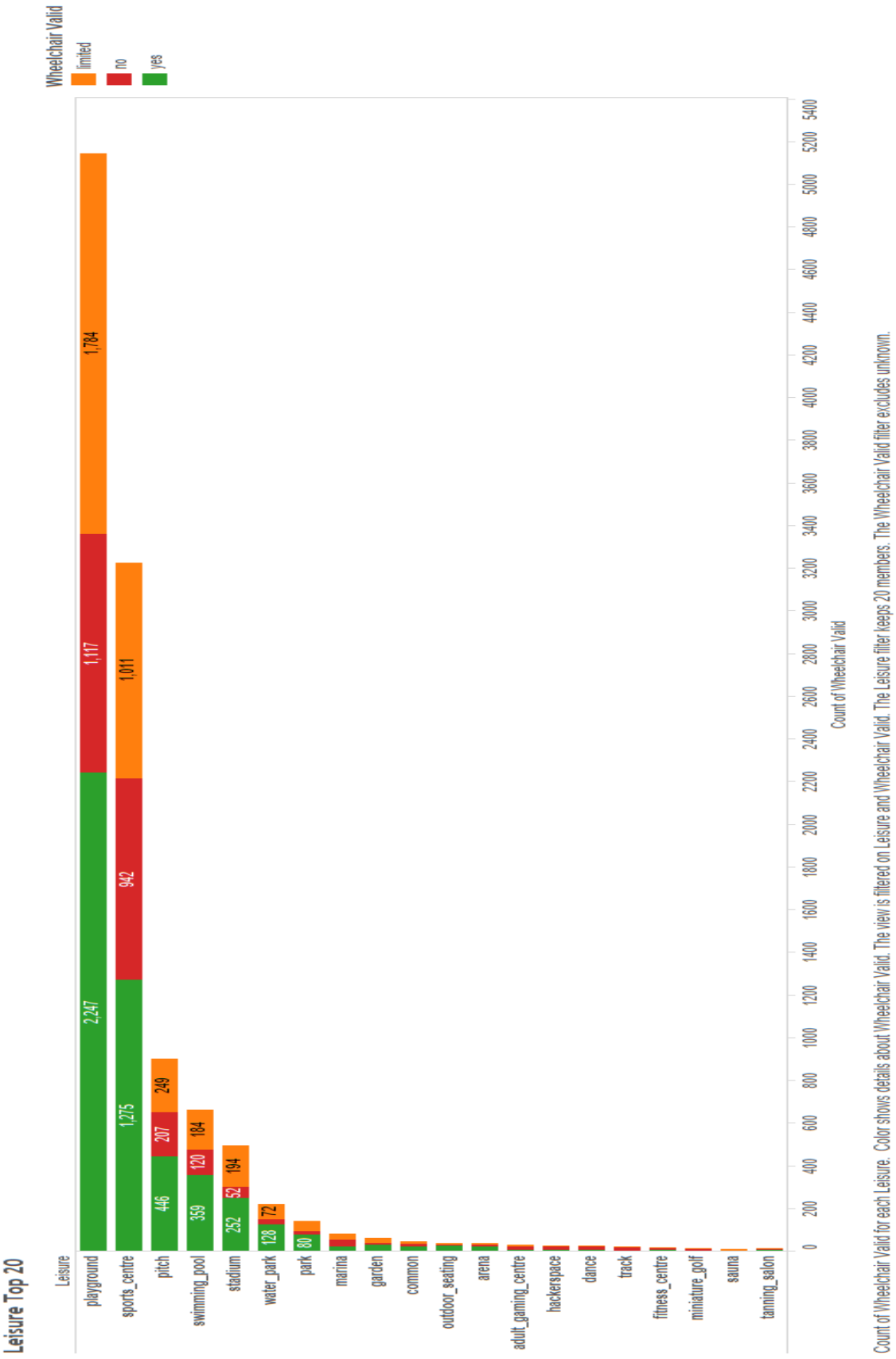

Figure 5.12: Values & Ranking Based On The Total POI count For The Variable *LEISURE*

Figure 5.13: Values & Ranking Based On The Total POI count For The Variable *PUBLIC_TRANSPORT*

hairdresser. Figure 5.14 displays the statistic for the same.

The basic amenities that make human lives comfortable are tagged under the *AMENITY* variable. Figure 5.15 shows the statistics for the variable and the mode for *AMENITY* is restaurant, while amenities like *townhall, hospital, social facility, public building* and *dentist* are right at the bottom.

5.2 Supervised Analysis

5.2.1 Classify if the node is known that is tagged or unknown that is not tagged

The class variable or the label column used for the binary classification analysis is *knownunknownflag*, which has the value *1* if the *NODE* is tagged and has the value *0* if the *NODE* is not tagged by the user. The variable *WHEELCHAIR_VALID* is removed from the list of variables for analysis because of the direct co-relation between the two. *NODEID* is a unique key, therefore, it is removed from the list of variables provided for analysis, thus provides no value for analysis.

Decision Trees

Decision Tree Classifier :

Maximum Depth	Impurity	Accuracy	Error
5	entropy	0.7933	0.1978
5	entropy	0.7928	0.2071
5	entropy	0.7923	0.2076
10	entropy	0.8143	0.1856

Random Forest :

Number of Trees	Maximum Depth	Impurity	Accuracy	Error
10	5	entropy	0.8010	0.1989
20	5	entropy	0.8004	0.1995
40	5	entropy	0.8143	0.1856

Gradient Boosted Trees :

Number of Trees	Maximum Depth	Impurity	Accuracy	Error
10	5	entropy	0.8209	0.1790
8004	5	entropy	0.8004	0.1995

Figure 5.14: Values & Ranking Based On The Total POI count For The Variable *SHOPS*

Figure 5.15: Values & Ranking Based On The Total POI count For The Variable *AMENITY*

Logistic Regression

elasticNetParam	regParam	Intercept	Accuracy	Error
0.9	0.8	-1.3405, 0.5105	0.7933	0.2066
0.5	0.5	-1.3417, 0.5102	0.7928	0.2071
0.5	0.8	-1.3431, 0.5099	0.7923	0.2076

Support Vector Machine

SVM-Type	SVM-Kernel	degree	gamma	coef.0	Accuracy	Error
C-classification	polynomial	10	8.446659e-05	10	0.6519	0.3480
C-classification	polynomial	10	8.446659e-05	6	0.5919	0.4080
C-classification	polynomial	5	8.446659e-05	6	0.8016	0.1983
C-classification	polynomial	6	8.446659e-05	5	0.5603	0.4390

The experimentation of results with the kernels except polynomial kernel resulted in the bias towards *class 0* as it is extremely high in compared to the *class 1*. Therefore experiments proceeded with polynomial kernel.

Neural Nets - Multilayer Perceptron

Layers	Accuracy	Error
[24, 50, 2]	0.7930	0.2069
[24, 10, 2]	0.7927	0.2072
[24, 50,10, 2]	0.7934	0.2065
[24, 10,10, 2]	0.7932	0.2067

5.2.2 Classify the class of the node as one of the wheelchair_valid class label i.e. yes, no or limited

The question discussed in the previous sub-section can be extended further to predict the class of the node if it is marked by the user i.e. yes, no or limited. Therefore the variable to classify is *WHEELCHAIR_VALID* with classes excluding the *unknown* class. The label column *knownunknownflag* is replaced by *WHEELCHAIR_VALID* excluding features with the class label *unknown*. The rest of the process of classification remains same as defined for the previous question.

Decision Tree

Decision Tree Classifier
The following experiments were run:

Maximum Depth	Impurity	Accuracy	Error
5	entropy	0.5558	0.4440
5	entropy	0.5361	0.4638
10	entropy	0.5401	0.4598

Random Forest
The following experiments were run:

Number of Trees	Maximum Depth	Impurity	Accuracy	Error
10	5	entropy	0.5381	0.4618
40	5	entropy	0.5361	0.4638
40	10	entropy	0.5401	0.4598

Support Vector Machine

SVM-Type	SVM-Kernel	degree	gamma	coef.0	Accuracy	Error
C-classification	polynomial	10	8.446659e-05	10	0.7753	0.2247
C-classification	polynomial	10	8.446659e-05	6	0.5919	0.4080
C-classification	polynomial	5	8.446659e-05	6	0.8016	0.1983
C-classification	polynomial	6	8.446659e-05	5	0.7814	0.2185

Neural Nets

Layers	Accuracy	Error
[24, 50, 2]	0.5362	0.4637
[24, 10, 2]	0.5345	0.4654
[24, 50,10, 2]	0.5363	0.4636
[24, 10,10, 2]	0.5355	0.4644

5.2.3 Supervised Analysis Overview & Suggestion

The performance of all algorithms is nearly the same. But *Support Vector Machine* for higher order polynomial kernel performs distinctly better to classify the labels correctly. Ensemble method, *Random Forest* works as good as the SVM. The ease of use and the ability to handle categorical variables with ease provides *Random Forest* algorithm the edge compared to high order Support Vector Machines, more over the time taken to train the model is more for SVM.

The supervised analysis provides good results and demonstrates that machine learning classification models can be used to classify or predict if the node label. This can be used by the campaign managers of the *CAP4Access*

initiative to plan. The campaign managers can target regions with a large number of available nodes that are predicted as tagged or avoid regions with a large number of available nodes that are predicted as not tagged. The Predictions or classifications can also be used to create targeted campaigns focused on areas that may contain a large number of selected node labels, for example, *yes* labels. This helps in running campaigns focused on getting more nodes or places which may be accessible or not accessible.

5.3 Unsupervised Analysis

Unsupervised learning is a type of machine learning algorithm used to draw inferences from datasets consisting of input data without labelled responses[37]. The most known examples of unsupervised learning are clustering and association rules.

5.3.1 Association Rules

Apriori

Apriori[1] is a well known algorithm for association rule data mining. The missing values of the nominal variables were replaced by '*not tagged*' because there is a possibility of an interesting relationship or pattern. Weka's Apriori implementation of algorithm helps us identify patterns. The variables and information for pattern are provided below:

1. WHEELCHAIR_VALID
2. TOILETS_WHEELCHAIR
3. SHOP
4. TOURISM
5. SPORT
6. PUBLIC_TRANSPORT
7. AMENITY
8. LEISURE
9. OFFICE
10. HISTORIC

11. AEROWAY
12. AERIALWAY
13. BUILDING
14. USERNAME

Best rules found :

1. AERIALWAY=not tagged ==> AEROWAY=not tagged
2. OFFICE=not tagged ==> AEROWAY=not tagged
3. OFFICE=not tagged, AERIALWAY=not tagged
   ==> AEROWAY=not tagged
4. TOILETS_WHEELCHAIR=not tagged
   ==> AEROWAY=not tagged
5. TOILETS_WHEELCHAIR=not tagged, AERIALWAY=not tagged
   ==> AEROWAY=not tagged
6. TOILETS_WHEELCHAIR=not tagged, OFFICE=not tagged
   ==> AEROWAY=not tagged
7. TOILETS_WHEELCHAIR=not tagged, OFFICE=not tagged, AERIALWAY=not tagged
   ==> AEROWAY=not tagged
8. HISTORIC=not tagged
   ==> AEROWAY=not tagged
9. HISTORIC=not tagged, AERIALWAY=not tagged
   ==> AEROWAY=not tagged
10. OFFICE=not tagged, HISTORIC=not tagged
    ==> AEROWAY=not tagged

The patterns are biased towards missing values that are *not tagged*, which is evident despite lowering of minimum support as the events are mutually exclusive therefore they are not interesting[5]. For example- if *historic* is *not tagged* then *aeroway* is also *not tagged*.

5.3.2 Clustering

Density Based Clustering

The experiments include the analysis of the tagged variables based on density. Figure 5.16 is the *DBSCAN* view of the variable *OFFICE* has the radius around core point(eps) is 0.4 and the minimum number of points to form the cluster is 20. The silhouette coefficient value is 0.2132. Figure 5.17 shows the high density clusters where tagging activity has taken place and the minimum sample used is 50. Figure 5.18 shows the density of leisure places being tagged by the users, the minimum sample used is 50 which delivers us two distinct clusters that can be seen in the figure. Figure 5.19 shows the density of wheelchair toilets being tagged by the users, the minimum sample used in the algorithm is 50. Figure 5.21 displays the density of tagging activity related to tourism , the silhouette coefficient is on the lower side. Figure 5.22 displays the density of tagging activity for places related to sport, the eps selected value is 0.3 and silhouette coefficient is 0.3024. Figure 5.20 displays the density of tagging activity for places related to amenities, the eps selected value is 0.3 and silhouette coefficient is 0.4230. Figure 5.23 displays the density of tagging activity for places related to the buildings, the eps selected value is 0.3 and silhouette coefficient is 0.3625.

All the density based clusters generated show the regions where the activity of tagging for an individual variable is high. In most of the graphics discussed above, the tagged nodes are concentrated around big cities of Germany. It is also worth noting that most of the clusters lie in west Germany.

K-Means

The analysis below tries to identify interesting clusters by application of the K-Means algorithm. The data utilized for the analysis excludes the *unknown* tagged rows as the analysis focuses on the tagging activity.

PERCENTAGE_HIGH_STATUS_HOUSEHOLDS & PERCENTAGE_OF_WOMEN

In the endeavour to explore the relationship between high status households and the number of women in the area, we perform the following analysis. Figure 5.24 shows 2 clusters separated by a boundary close to 40% PERCENTAGE_HIGH_STATUS_HOUSEHOLD. The plot is parallel to the y-axis. The variation of percentage of women in the postal areas where the node is tagged has a low variance. With the exception of one point on the extreme right

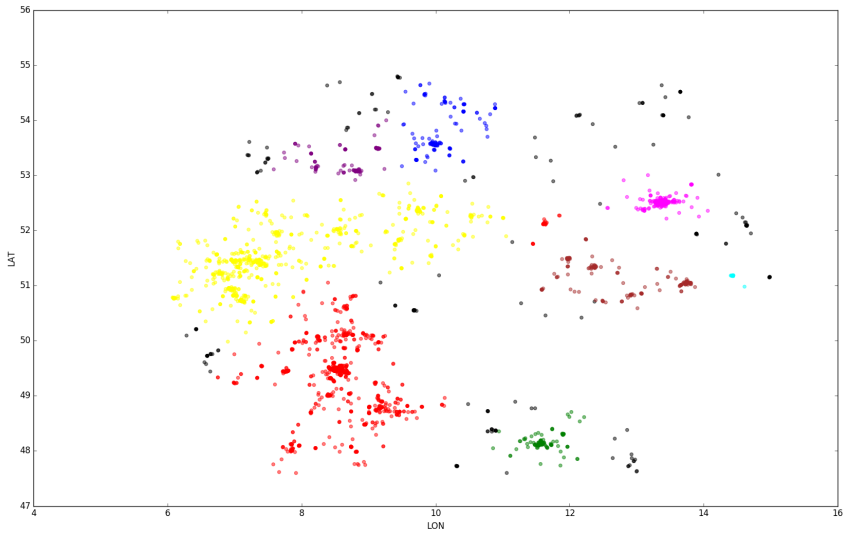

Figure 5.16: Density Based Clustering with eps=0.4, silhouette=0.2245 & variable *OFFICE*

with 100% high status household where the percentage of women is around 65%, there is nothing too significant.

PERCENTAGE_LOW_STATUS_HOUSEHOLDS & PERCENTAGE_OF_WOMEN

Figure 5.25 separates the data into 2 clusters at the same level as discussed in the previous section and the plot is parallel to the y-axis, with almost the same variance. There is no interesting pattern in the graphics.

PERCENTAGE_HIGH_STATUS_HOUSEHOLDS & PERCENTAGE_OF_MEN

Figure 5.26 compares the tagging activity with respect to the men. The results are the same as that of the women and the separation of the clusters is at around 40% value of the variable *PERCENTAGE_HIGH_STATUS_HOUSEHOLDS*. Therefore, all the places tagged are in the places where the percentage of men varies from 40% to 50%.

NODEIDCOUNT VS NUMBEROFHOUSEHOLDS

Figure 5.27 shows a strong correlation between the count of tagged *NODES* and the total number of households in the postal area. The 3 clusters formed show that as the count and the number of households increase either on

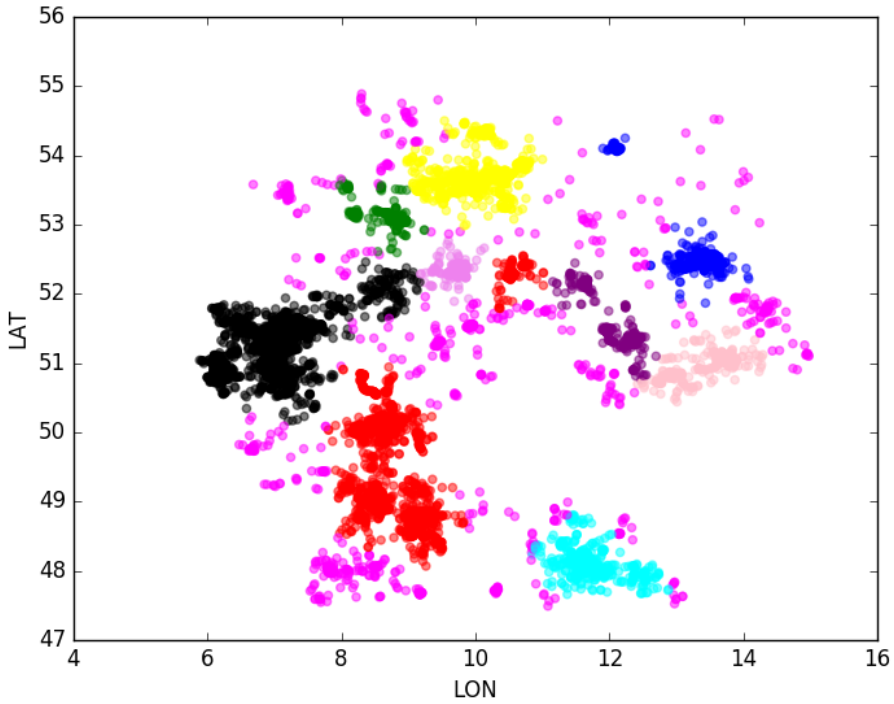

Figure 5.17: Density Based Clustering with eps=0.3, silhouette=0.5048 & variable *PUBLIC_TRANSPORT* excludes value '*not tagged*'

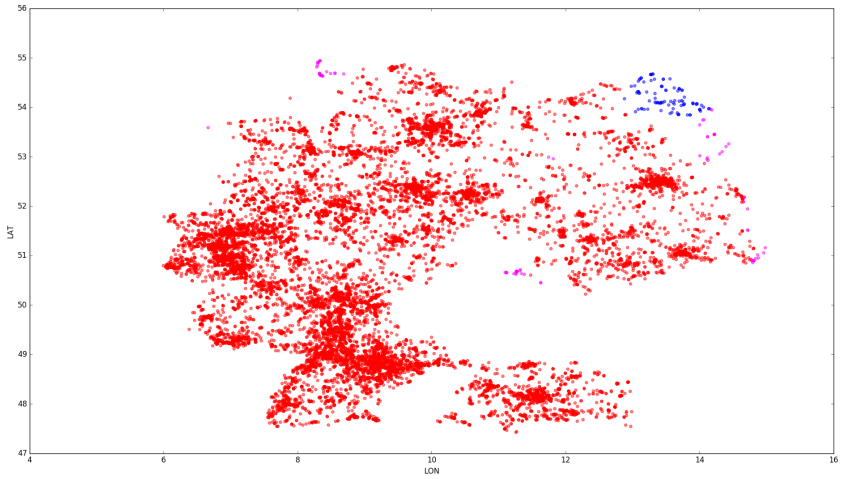

Figure 5.18: Density Based Clustering with eps=0.3, silhouette=0.2429 & *LEISURE* excludes value 'not tagged'

the x-axis or the y-axis the variance increases and the intra-cluster distance decreases. The cluster in blue is a sparse cluster with postal areas with the higher number of households.

NODEIDCOUNT & PLZ_EWA_A_GESAMT
Figure 5.28 also complements the previous analysis with the total number of households. The same is observed with the total population as the is a very strong correlation between the total number of households and the total population. Therefore, the graph looks very similar.

NODEIDCOUNT & NUMBEROFCOMMERCIALBUILDINGS
The analysis explores the possible relationship between the number of buildings that are used commercially to the node count in the area. Figure 5.29 shows the analysis between the number of commercial buildings in the postal code and the total number of *NODES*. The plot is almost parallel to the x-axis. The clusters separate the plots as bins with increasing variance along the x-axis.

NODEIDCOUNT & PERCENTAGE_AVG_STATUS_HOUSEHOLDS

Figure 5.30 plots the relationship between the total number of tagged *NODES*

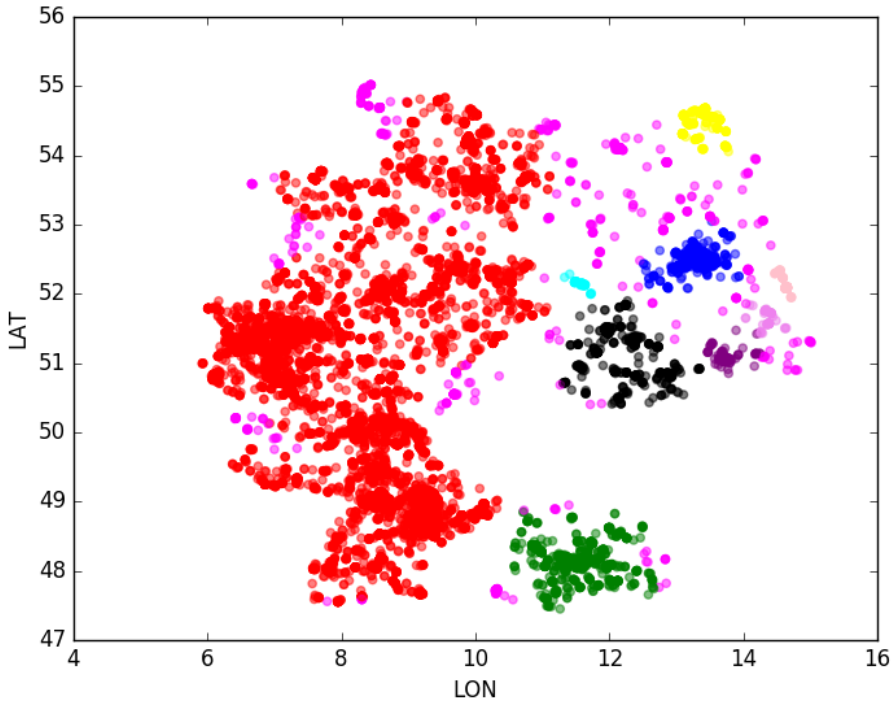

Figure 5.19: Density Based Clustering with eps=0.3, silhouette=0.2029 & variable *TOILETS_WHEELCHAIR* excludes value 'not tagged'

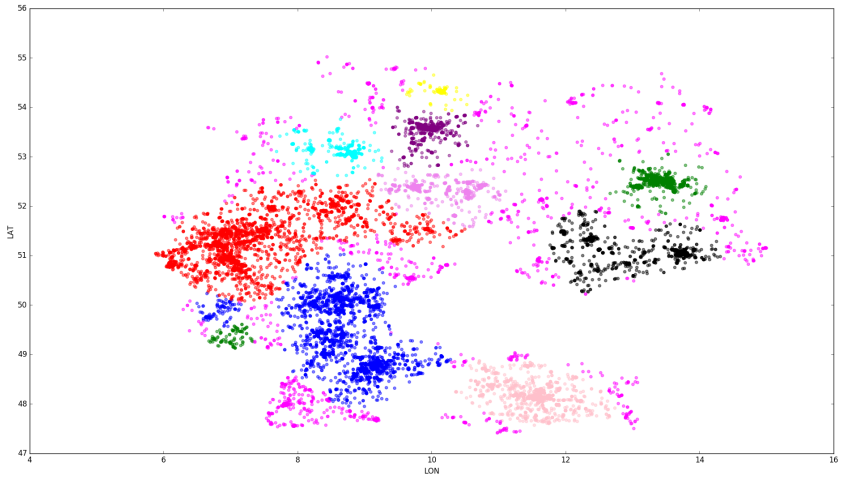

Figure 5.20: Density Based Clustering with eps=0.3, silhouette=0.4230 & variable *BUILDING* & excludes value 'not tagged'

and the percentage of average income households. The plot is almost parallel to x-axis and the clusters created demonstrate variance.

NODEIDCOUNT & PERCENTAGE_OF_MEN, & NODEIDCOUNT & PERCENTAGE_OF_WOMEN

Figure 5.31 analyses the relation between the tag count to the percentage of men in a postal area. The analysis is similar to Figure 5.26. Figure 5.32 analyses the relation between the tag count to the percentage of men in a postal area and the area is also similar to the previous analysis in Figure 5.24.

5.3.3 Unsupervised Analysis Overview & Suggestion

Association Rules

Direct query on the database shows a subset with the interesting combination of transactions for a node. For example, a user tags a node with the variable *PUBLIC_TRANSPORT=stop_position* and *HIGHWAY=bus_stop* at the same time. Another example is that of variable *TOURISM=artwork*, *HISTORIC=memorial* and *SPORT=yes*. Thus, an interest in the exploration of such patterns was evident. Although the association rule algorithms did not return expected results due to the high affinity in the table towards *not*

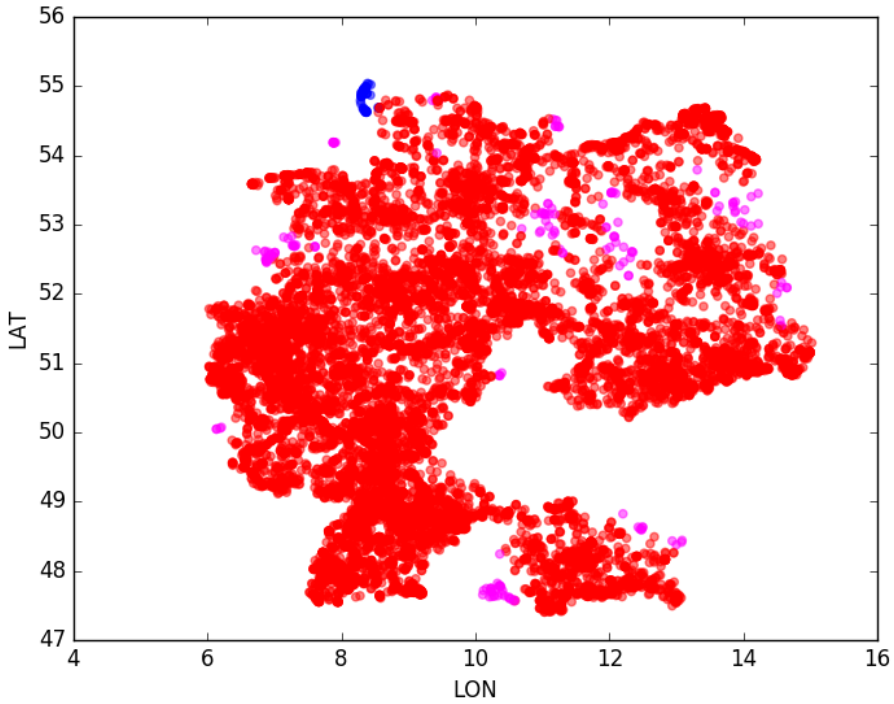

Figure 5.21: Density Based Clustering with eps=0.3, silhouette=0.054323 & variable *TOURISM* excludes value 'not tagged'

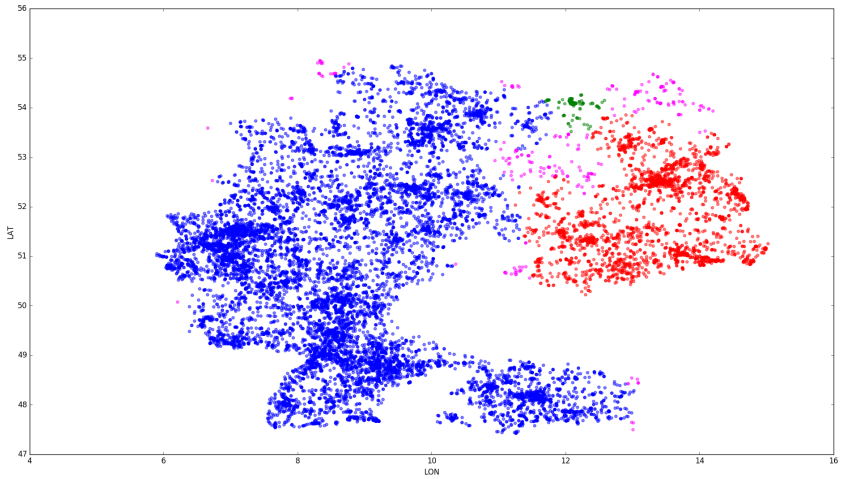

Figure 5.22: Density Based Clustering with eps=0.3, silhouette=0.3024 & variable *SPORT* excludes value 'not tagged'

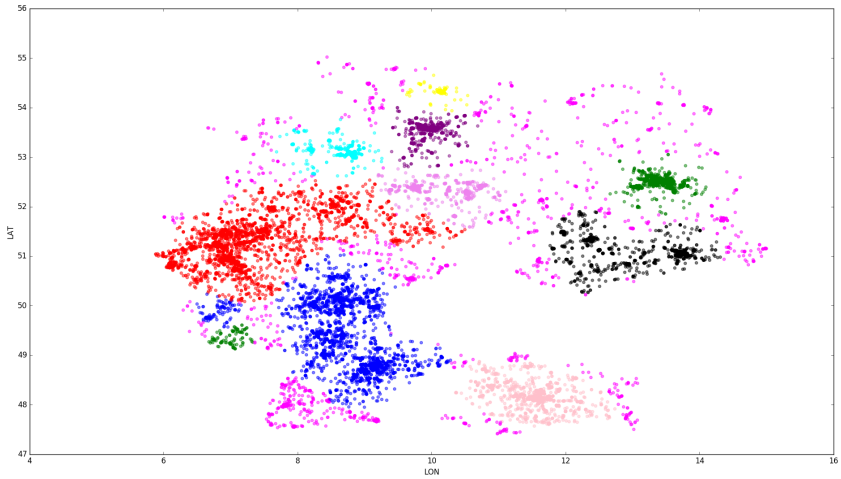

Figure 5.23: Density Based Clustering with eps=0.3, silhouette=0.4230 & variable *BUILDING* excludes value 'not tagged'

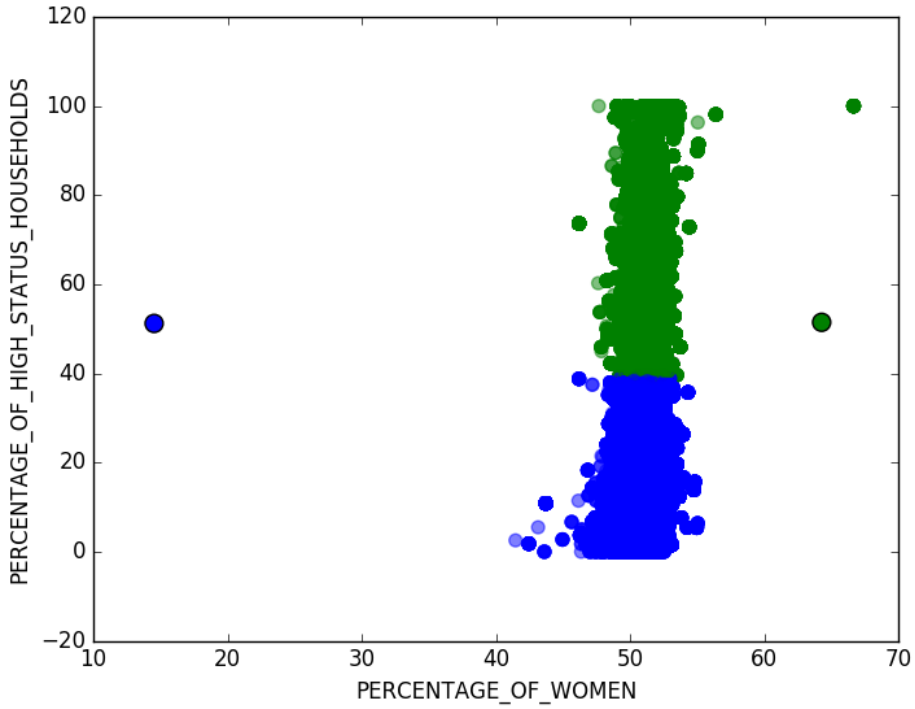

Figure 5.24: KMeans with n_cluster=2 & silhouette=0.0230, *PERCENTAGE_OF_WOMEN* to *PERCENTAGE_HIGH_STATUS_HOUSEHOLD*

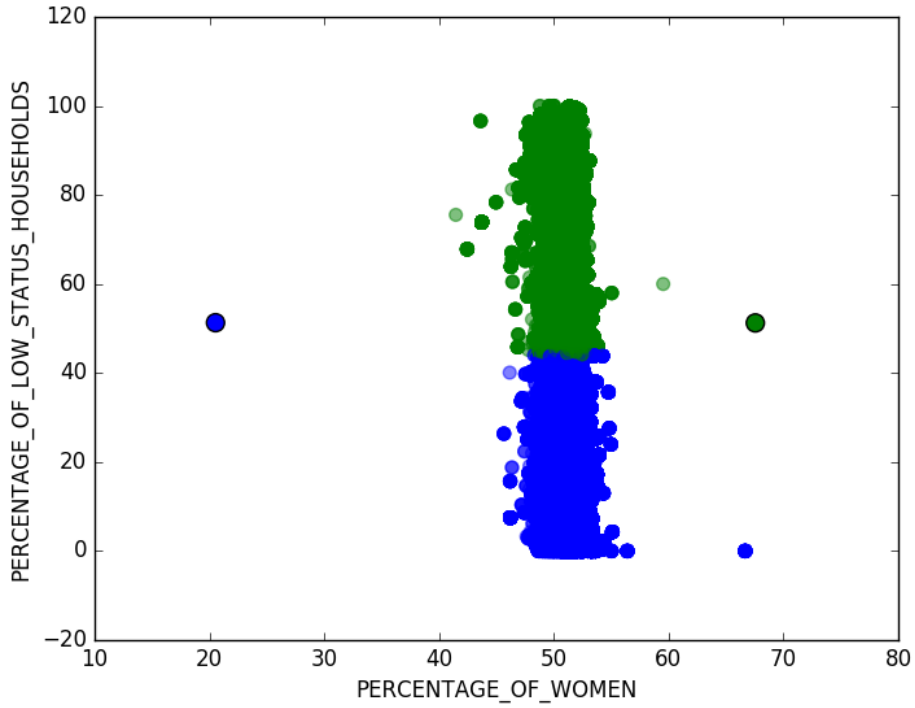

Figure 5.25: KMeans with n_cluster=2 & silhouette=0.0914, *PERCENTAGE_OF_WOMEN* to *PERCENTAGE_LOW_STATUS_HOUSEHOLDS*

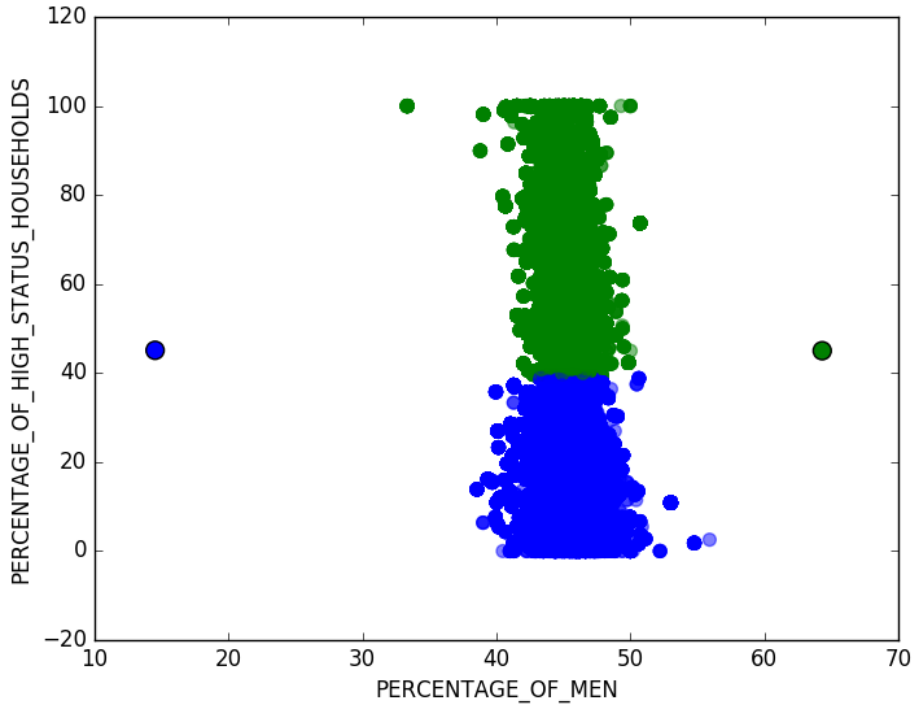

Figure 5.26: KMeans with n_cluster=2 & silhouette=0.1618, *PERCENTAGE_HIGH_STATUS_HOUSEHOLDS* to *PERCENTAGE_OF_MEN*

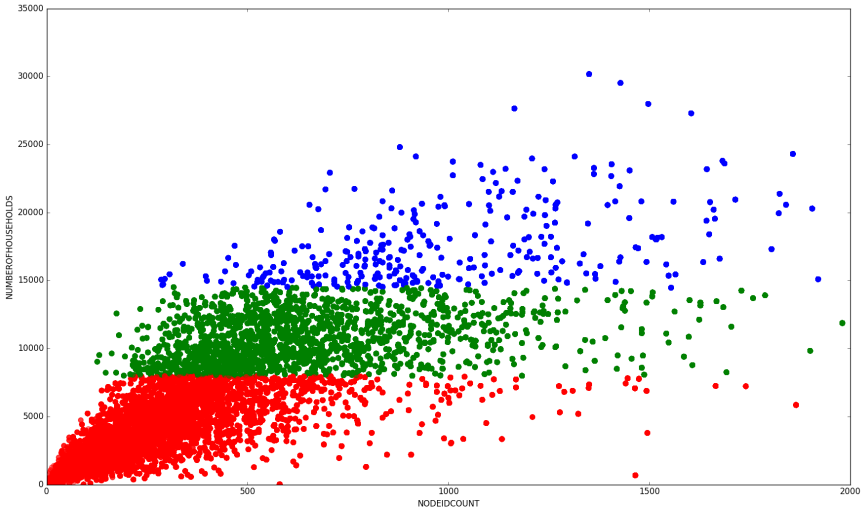

Figure 5.27: KMeans with n_cluster=3 & silhouette=0.2011, *NODEIDCOUNT* to *NUMBEROFHOUSEHOLDS*

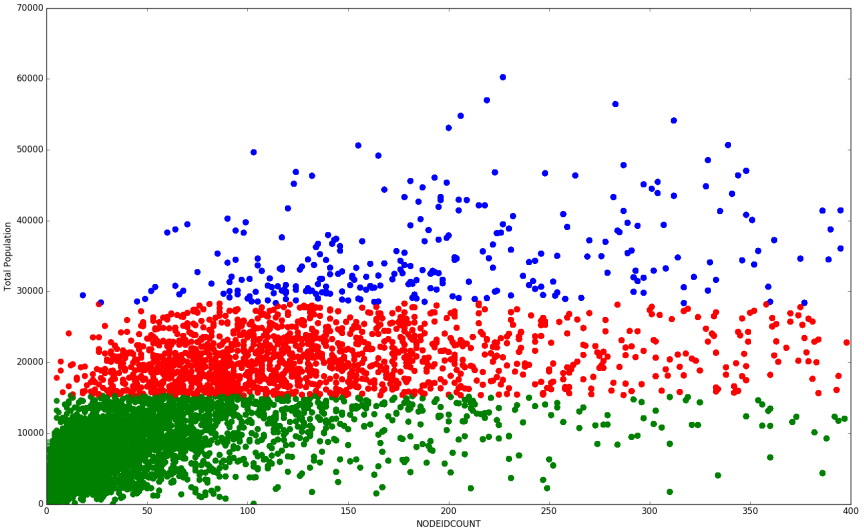

Figure 5.28: KMeans with n_cluster=3 & silhouette=0.0614, *NODEIDCOUNT* to *PLZ_EWA_A_GESAMT(Total Population)*

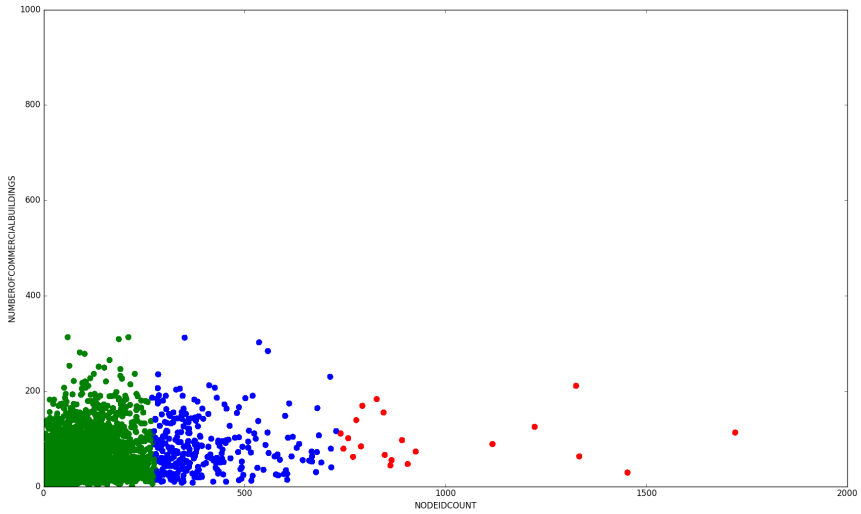

Figure 5.29: KMeans with n_cluster=3 & silhouette=0.1926, *NODEIDCOUNT* to *NUMBEROFCOMMERCIALBUILDINGS*

tagged values, still it remains extremely lucrative for analysis.

Clustering

The scope of analysis for density based clustering was to identify the dense regions of tagged places. The tagging activity of users is targeted mostly in big cities and the focus needs to be shifted towards less dense regions that is in the towns and villages. The application of the K-Means algorithm was to identify groups that show unusual behaviour. The analysis shows that the German gender ratio is balanced. It is also worth noting that an unusual cluster is observed when the node count exceeds 1000 for a postal area in Figure 5.31 and Figure 5.32. There was nothing significant that was observed as part of the unsupervised analysis.

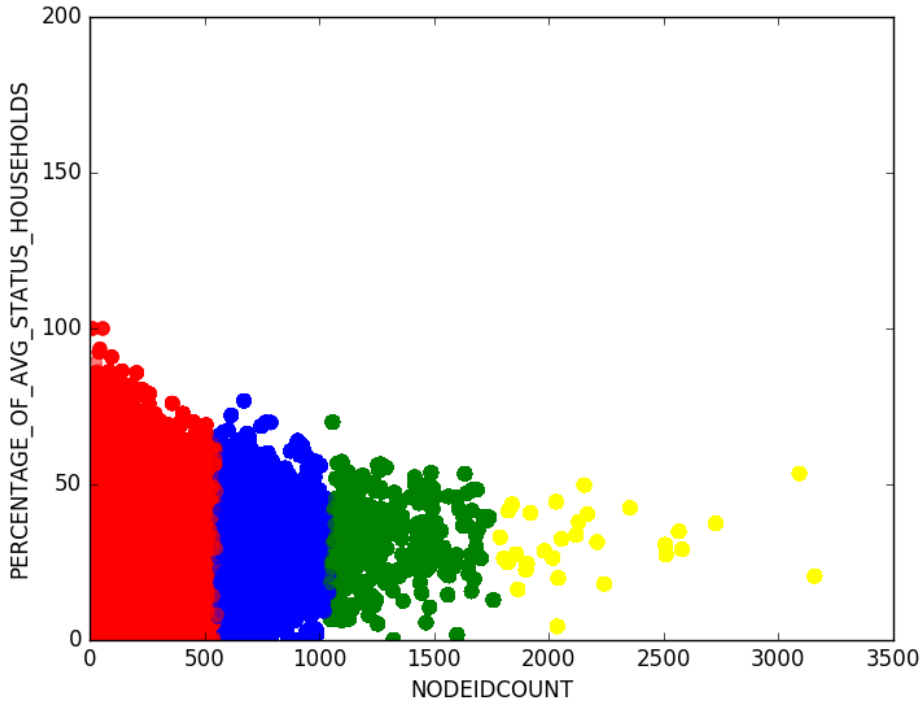

Figure 5.30: KMeans with n_cluster=4 & silhouette=0.119, *NODEIDCOUNT* to *PERCENTAGE AVG STATUS HOUSEHOLDS*

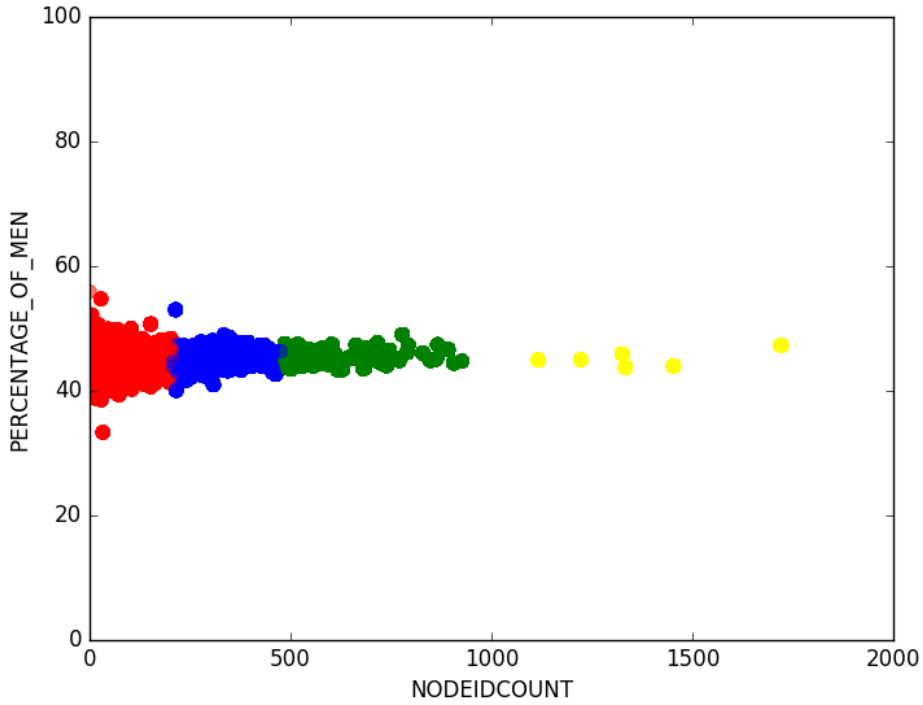

Figure 5.31: K-Means with n_cluster=3 & silhouette=0.0214, *NODEIDCOUNT* to *PERCENTAGE OF MEN*

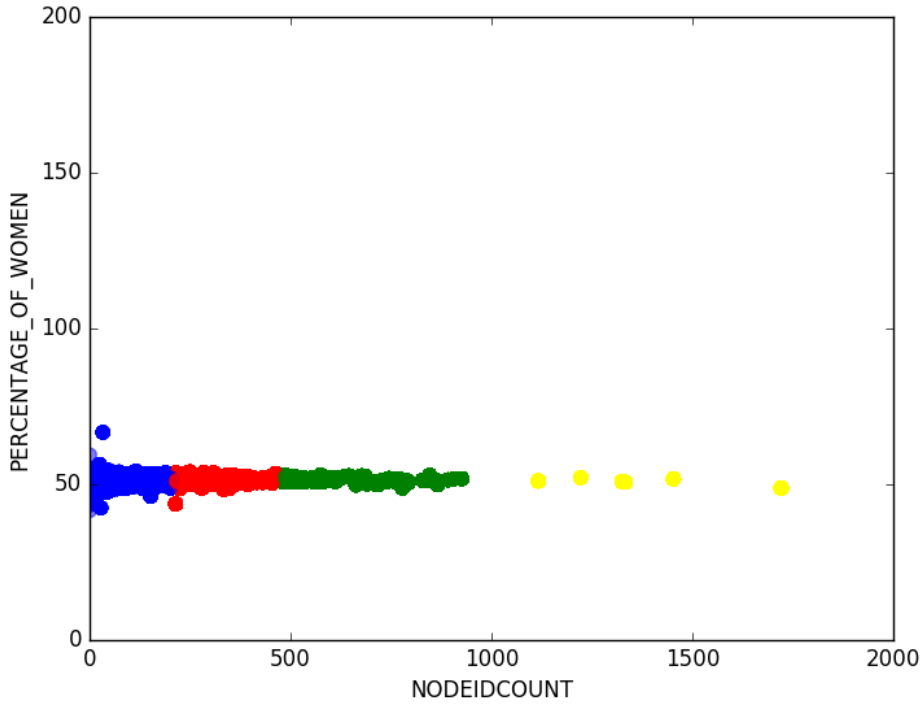

Figure 5.32: KMeans with n_cluster=4 & silhouette=0.3543, *NODEIDCOUNT* to *PERCENTAGE OF WOMEN*

Chapter 6

Summary, Conclusion And Future Works

6.1 Summary & Conclusion

The topic for the thesis originated from the *CAP4ACCESS* project run by the European Commission and its partners, which deals towards the sensitization of people and development of tools for awareness about people with movement disabilities.

The explorative analysis is never ending and to explore and find interesting patterns and the results is a tedious task. Therefore, a scientific approach was very important. To start with, familiarizing the domain and the data sources were done. Thereafter, selection of methodology for data analysis was done which resulted in the use of *CRISP-DM* methodology. The data sources are the source of blood to the analysis methodology, and as there were two sources of data that is *MICROM* and *OSM Wheelchair History(OWH)*, it was important to integrate them together to extract relevant datasets. Therefore a functional and technically impure data warehouse was created, from which the datasets are extracted and analysed.

The next task was to select appropriate tools for analysis. This task was very important as the data set although was not big data but contained a large number of rows. After careful analysis, *Apache spark* and its machine learning library were utilized for building and testing supervised models. *DataFrame* API for *Python*, *Pandas*, the machine learning library *Sci-kit learn* provided unsupervised algorithms for analysis, the association rule analysis was performed using *WEKA*. *Tableau*[21] and *Matplotlib*[24] provide attractive visualizations for representation and analysis.

6.2 Future Works

The scope of future works is immense. The data integration provides a workable solution for the thesis. It can be extended to a complete data warehouse using Kimball's[26] methodology. This can help in easy integration, maintenance and utilization of data for efficient analysis at a commercial scale. Furthermore, the analysis can be extended by the formulation of more analytic questions from other domains apart from social intelligence and make a more market oriented analysis on the extraction of results from the data by inclusion of more variables. More analysis can be extended by applying new machine learning algorithms like *stochastic gradient decent*[35], *Naive Bayes*[40], etc. The results can then be compared to the results of the thesis. The unsupervised analysis done focuses on identification of unusual patterns in the data, but the variables selected for analysis show no interesting trends or patterns. Due to limited time of the thesis the scope of unsupervised analysis was restricted to identification of the interesting patterns and clusters, this can be extended to identification and explanation of the occurring pattern and behaviour. Visualization of unsupervised analysis can be improved by using *Folium*[9], the geo-spatial python library. From a technical perspective, an incremental automation of analysis is possible with the extension of the thesis, that may help researchers, select right technologies and save time on handling the huge data.

Bibliography

[1] Rakesh Agrawal, Ramakrishnan Srikant, et al. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487–499, 1994.

[2] S Aranganayagi and K Thangavel. Clustering categorical data using silhouette coefficient as a relocating measure. In *Conference on Computational Intelligence and Multimedia Applications, 2007. International Conference on*, volume 2, pages 13–17. IEEE, 2007.

[3] Kristin P Bennett and Colin Campbell. Support vector machines: hype or hallelujah? *ACM SIGKDD Explorations Newsletter*, 2(2):1–13, 2000.

[4] B Borah and DK Bhattacharyya. An improved sampling-based dbscan for large spatial databases. In *Intelligent Sensing and Information Processing, 2004. Proceedings of International Conference on*, pages 92–96. IEEE, 2004.

[5] Christian Borgelt and Rudolf Kruse. Induction of association rules: Apriori implementation. In *Compstat*, pages 395–400. Springer, 2002.

[6] Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pages 161–168. ACM, 2006.

[7] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[8] Statistisches Bundesamt Deutschland. "Gebiet und Bevlkerung Flche und Bevlkerung" (in German). `http://www.statistik-portal.de/Statistik-Portal/de_jb01_jahrtab1.asp`. [Online; accessed; 5 August 2014].

[9] Michael Diener. *Python Geospatial Analysis Cookbook*. Packt Publishing Ltd, 2015.

[10] Evgenia Dimitriadou, Kurt Hornik, Friedrich Leisch, David Meyer, and Andreas Weingessel. Misc functions of the department of statistics (e1071), tu wien. *R package*, 1:5–24, 2008.

[11] Fraunhofer Institute for Intelligent Analysis & Information Systems MappingForChange Polibienestar Research Institute University of Valencia Sozialhelden e.V. University of Heidelberg University College London (UCL) Zentrum fr Soziale Innovation empirica Gesellschaft fr Kommunikations-und Technologieforschung mbH, Elche City Council. Welcome to cap4access. `http://www.cap4access.eu/intro/`. [Online; accessed].

[12] UN Enable. Factsheet on persons with disabilities. *Nueva York: United Nations. Última consulta*, 15(11):2013, 2008.

[13] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37, 1996.

[14] Apache Software Foundation. SparkML Decision Trees. `http://spark.apache.org/docs/latest/ml-classification-regression.html#decision-trees/`. [Online; accessed].

[15] Apache Software Foundation. SparkML GBTrees. `http://spark.apache.org/docs/latest/mllib-ensembles.html#gradient-boosted-trees-gbts`. [Online; accessed].

[16] Apache Software Foundation. SparkML logistic-regression. `http://spark.apache.org/docs/latest/ml-classification-regression.html#logistic-regression`. [Online; accessed].

[17] Apache Software Foundation. SparkML Multilayer Perceptron. `http://spark.apache.org/docs/latest/ml-classification-regression.html#multilayer-perceptron-classifier`. [Online; accessed].

[18] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

[19] Mordechai Haklay and Patrick Weber. Openstreetmap: User-generated street maps. *Pervasive Computing, IEEE*, 7(4):12–18, 2008.

[20] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *Unsupervised learning*. Springer, 2009.

[21] Jeffrey Heer, Jock D Mackinlay, Chris Stolte, and Maneesh Agrawala. Graphical histories for visualization: Supporting analysis, communication, and evaluation. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1189–1196, 2008.

[22] Geoffrey Holmes, Andrew Donkin, and Ian H Witten. Weka: A machine learning workbench. In *Intelligent Information Systems, 1994. Proceedings of the 1994 Second Australian and New Zealand Conference on*, pages 357–361. IEEE, 1994.

[23] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007.

[24] John D Hunter et al. Matplotlib: A 2d graphics environment. *Computing in science and engineering*, 9(3):90–95, 2007.

[25] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[26] Ralph Kimball. *The data warehouse lifecycle toolkit: expert methods for designing, developing, and deploying data warehouses*. John Wiley & Sons, 1998.

[27] Ralph Kimball, Margy Ross, et al. The data warehouse toolkit: the complete guide to dimensional modelling. *US: John Wiley & Sons*, 2002.

[28] David G Kleinbaum and Mitchel Klein. *Logistic regression: a self-learning text*. Springer Science & Business Media, 2010.

[29] Ron Kohavi. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *KDD*, volume 96, pages 202–207. Citeseer, 1996.

[30] Vineet Kumar and Werner Reinartz. *Customer relationship management: Concept, strategy, and tools*. Springer Science & Business Media, 2012.

[31] Mitchell P LaPlante et al. Assistive technology devices and home accessibility features: prevalence, payment, need, and trends. *Advance data from vital and health statistics*, 1992.

[32] Stephen J. Macdonald and John Clayton. Back to the future, disability and the digital divide. *Disability & Society*, 28(5):702–718, 2013.

[33] Xiangrui Meng, Joseph Bradley, Burak Yavuz, Evan Sparks, Shivaram Venkataraman, Davies Liu, Jeremy Freeman, DB Tsai, Manish Amde, Sean Owen, et al. Mllib: Machine learning in apache spark. *arXiv preprint arXiv:1505.06807*, 2015.

[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[35] Shai Shalev-Shwartz and Ambuj Tewari. Stochastic methods for l 1-regularized loss minimization. *The Journal of Machine Learning Research*, 12:1865–1892, 2011.

[36] Pang-Ning Tan, Michael Steinbach, Vipin Kumar, et al. *Introduction to data mining*, volume 1. Pearson Addison Wesley Boston, 2006.

[37] Inc The MathWorks. mathworks unsupervised learning. `http://de.mathworks.com/discovery/unsupervised-learning.html`. [Online; accessed].

[38] James J Thomas and Kristin A Cook. A visual analytics agenda. *Computer Graphics and Applications, IEEE*, 26(1):10–13, 2006.

[39] Sergei Vassilvitskii and David Arthur. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035, 2006.

[40] Zhou Wang, Hongjian Fan, and Kotagiri Ramamohanarao. Exploiting maximal emerging patterns for classification. In *AI 2004: Advances in Artificial Intelligence*, pages 1062–1068. Springer, 2004.

[41] Ying Zhao and George Karypis. Evaluation of hierarchical clustering algorithms for document datasets. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 515–524. ACM, 2002.